

Game Design & Dev - Planning - Formal Elements

- Dr Nick Hayward

A brief intro to a consideration of formal elements for initial game design and development.

Contents

- Intro
- Players
 - players, patterns, and numbers
- Importance of objectives
- A consideration of procedure
 - procedures in development
- Rules and game concepts
- Rules, objects, and updates
- References

Intro

We may often define formal structure for a game as comprising the following,

- players
- objectives
- procedures & rules
 - including implied **boundaries**
- conflict, challenge, battle...
- outcome, end result...

These constituent elements come together to form what we largely understand to be a game. Such formal elements constitute how we design, structure, and develop our video games. The overlap and interplay of these formal elements has now become the foundation for game design.

A sound understanding and knowledge of these formal elements, their usage, and application, allows us as designers and developers to start creating innovative, playful game experiences.

Players

For a given game, we may identify the need for rules, procedures, and so on, However, within the confines of such rules, players are suspending normal societal restrictions to enact shooting, fighting, role-play, and magical roles normally confined to a passive medium such as literature and film.

Such actions can often form stark contrasts in a game environment. We destroy an alien ship to save the world, thereby killing all of the aliens but saving the humans. Such rules become enacted in a **magic circle**, as originally described by Huizinga in his seminal 1938 title, **Homo Ludens**. It was later adapted, and refashioned for digital games, by Salen and Zimmerman in their title, *Rules of Play*,

In a very basic sense, the magic circle of a game is where the game takes place. To play a game means entering into a magic circle, or perhaps creating one as a game begins.

Salen, K. & Zimmerman, E. *Rules of Play: Game Design Fundamentals*. MIT Press. 2003.

Such rules, within the defined confines of a game, naturally create the opportunity for play.

By our use of rules, characters, story, and even mechanics and control, we invite a player to become involved with, and invested in, our game. Motion controllers, including Nintendo's Wii, Microsoft's Xbox Kinect, and Sony's Playstation Move, became an invitation for players to intuitively enter a game. The premise of many games has now become an extension of the controller, for example.

It's not only a matter of engaging and inviting players into your game. You'll also need to consider the nature and structure of player participation. For example, how many players does the game support? Will each player adopt the same role? Perhaps in a team or in direct competition.

How we answer such questions will have a direct influence on the nature of the game we're designing, its gameplay, and a player's engagement with the story and characters.

players, patterns, and numbers

Games may be designed and developed for a variety of player numbers, from strict single player options to varied multiplayer environments. Massively multiplayer online games push player numbers to the upper bounds. They provide playable environments for single players to engage individually with a game or simply communicate, collaborate, and compete against tens of thousands of other players.

However, your choice of player numbers will often determine patterns of interaction for your game as well. Such patterns may often include play against the machine (the game itself), against another player, collaborative play, and play against many players including group play. For example

- single player versus the game
 - e.g. Space Invaders, Mario, Sonic...most platformers &c.
- multiple individual players versus the game
 - e.g. sports, racing, card games...
- single player versus another player
 - draughts (checkers), boxing and fighting games such as Street Fighter and Mortal Kombat...
- multiple players versus a single player
 - many detective and role playing board games include such features
 - some god games may also be structured using this pattern
- collaborative play
 - players work together against the game
 - again, sports games such as tennis doubles
 - other examples such as *Journey* by designer Jenova Chen...
- multiple players competing
 - e.g. Halo, Call of Duty...
 - standard pattern referenced for *multiplayer* games
- team competitions...
 - e.g. eSports such as League of Legends...

Importance of objectives

Objectives may help establish different requirements and goals in a game, thereby helping a user to achieve results within the confines of the rules of the game. These may seem challenging and difficult, but when correctly designed relative to such rules these objectives should also seem achievable to a player.

A game's objective may also help set the tone for gameplay and interaction. The objective of most platform games, for

example, is quite different from a sports based game. The tone for each of these games becomes a reflection of the objective.

The use of objectives in games is not limited to just the overall game itself. We may consider defining an objective for different player roles, or perhaps as mini challenges within our games. Each level may define its own objective, for example, such as completing a level as fast as possible, or collecting all available credit (coins, for example) on another. Our choice of such objectives needs to be considered carefully, and mixed appropriately, as each will affect not only the formal system of our game, but also the dramatic aspect. Good integration of objectives in the premise or story of a game helps strengthen dramatic aspects as well. Legend of Zelda is a great example.

A consideration of procedure

As we start to define procedures for our game, we may start to see a few common actions that exist across multiple genres. These often include the following:

- an action to start the game - specific procedure required to initiate gameplay...
- ongoing actions and procedures - e.g. common, persistent actions that continue, repeat &c. as part of the game
- reserved or special actions - e.g. actions that may be required and executed due to a given condition or game requirement
- actions to conclude or resolve - e.g. resolve actions at certain points within the game, or at the end of the game itself...

For video games, such actions and procedures will be closely associated with the way the player interacts with the game. For example, given key combinations or controller buttons - perhaps tapping particular options on the screen itself, or moving a mobile device to control certain actions.

Obviously, different controls, buttons, and device actions should also have separate actions in the game. Be careful not to change and confuse the player by modifying such actions and procedures.

Again, if we consider Super Mario Bros. we may clearly identify controls for given actions and procedures. For example, the expected usage for directional buttons, the option to jump or swim with the **A** button, &c.

procedures in development

Procedures also play a key role in the way we develop our games. We can add procedures within the logic of our game to monitor certain ongoing states, user interaction, updates, and rendering.

Effectively, these procedures are working in the core of our game, responding to changes in state. For example, a player completes a puzzle within the main game. We need to monitor the ongoing puzzle's responses, check the player input and interactions, and then update the state of the game in response to a success or failure result.

So, we're effectively checking whether a given action succeeds or not. Then, we determine the impact this may have on the game itself.

It's also important to realise that such procedures and actions are naturally limited by real-world constraints. These might include the performance of the underlying system, controllers, interaction options, the screen, and so on. Also, we may need to tailor such requirements to match the type of game we're developing, and the target audience.

A game developed for a general audience on a mobile device may require different procedures and actions to an educational game targeted to an online, keyboard/mouse combination.

Rules and game concepts

As we define and formalise rules for our games, we need to consider more than simply the gameplay itself.

Objects within our games, and concepts embedded in gameplay structures, require defined limitations and rules. Such game objects, including, for example, characters, weapons, vehicles, obstacles, and so on, may be derived or inspired by real-world objects. Such objects may, therefore, come with the perception of existing limitations and rules. A player

knows what these objects can and cannot do in the real world.

However, whilst these examples may be grounded in the real-world, we may simply use them as inspiration and starting points for our game's objects. We are not inherently limited or defined by them, and may modify as befits the requirements of our game, and its gameplay.

So, the context of the game will be a determining factor in the development of our game's objects. These objects may also be developed as a group of properties and variables, which together form the whole.

For example, in a world of chivalry, knights, ogres, and other fantastical creatures, we may still create concepts and objects that unify these characters. From base objects, we can simply inherit and modify as needed.

For example, we may require various characters to ride on horseback, or perhaps astride an elephant, or even a fictional dragon &c. Our objects may be abstracted to include known attributes, which can then be used as the parent for multiple real and imagined objects and characters within our game.

Rules, objects, and updates

As developers and designers, we need to ensure a balance between maintaining game objects and variables, and creating an intuitive update for our users. It's unlikely our player will want to keep a manual tally of such updates, so we need to consider how we may allow them to quickly and easily intuit game objects.

For example, we need to maintain a running total of game objects, such as coins, lives, energy levels, and so on, whilst correctly informing the player of any updates. Our player should be able to quickly learn the nature of these objects. If they're too difficult or complex, we need to consider how this affects our player's gaming experience.

We also need to ensure that there is sufficient isolation between different objects, or that at least our player can discern differences without too much effort or guesswork.

Such updates may also be influenced by known restrictions in the game's rules. Such rules are useful in many respects, for example relative to boundaries, objectives, and objects themselves. By establishing rules, for example, to restrict objects and their attributes, we create a known scale for state within our game. This is particularly useful for both the designer/developer and, of course, the player.

The player has defined restrictions, they know what they can and can't do. Risk and reward is set in the game's logic and gameplay.

References

- Bogost, I. *Persuasive Games: The Expressive Power of Videogames*. MIT Press. Cambridge, MA. 2007.
- Bogost, I, *The Rhetoric of Video Games*. in *The Ecology of Games...* Salen, E. MIT Press. Cambridge, MA. 2008.
- Bogost, I. *Unit Operations: An Approach to Videogame Criticism*. MIT Press. Cambridge, MA. 2006.
- Huizinga, J. *Homo Ludens: A Study of the Play-Element in Culture*. Angelico Press. 2016.
- Murray, J. *Hamlet on the Holodeck: The Future of Narrative in Cyberspace*. Free Press. New York. 1997.
- Salen, K. & Zimmerman, E. *Rules of Play: Game Design Fundamentals*. MIT Press. 2003.