What Does Bach Have in Common with World 1-1: Automatic Platformer Gestalt Analysis

Johnathan Pagnutti
1156 High St
Santa Cruz, CA
jpagnutt@ucsc.edu

Abstract

Platformer level generation has often used a beat metaphor to relate to how players interact with level geometry. However, this conceptualization of beats is different from the musical concept of 'beat', limiting the utility of theories and tools developed in music analysis for platformer levels. A gameplay gestalt, a pattern of interaction that the player enacts or performs in order to make progress in a game, may fit the beat metaphor. By taking a very similar lens and viewing players playing platformer levels as enacting a series of gameplay gestalts through time, gestalt music analysis (GMA) does fit into the platformer domain. This paper details work on transforming a GMA model to work with the Platformer Experience Dataset (PED), and some promising first results of the transformed model.

Platformer level design and analysis has often flirted with musical concepts of rhythm and beat. As described in (Csikszentmihalyi 1990), platformers allow players to sink into a rhythm or 'flow' state where they are rapidly making distance calculations and keeping exact timing on inputs to progress forward. Metaphorically considering single obstacles or player actions as a 'beat' in time has been very productive for platformer level generation. Both (Compton and Mateas 2006) and (Smith, Cha, and Whitehead 2008) derive a concept of rhythm from beats, where beats correspond to player actions to progress past level features.

It does not seem beyond the pale to want to take the platformer idea of beat and see how well it fits in with a musical idea of beat. This would allow us to use techniques for music analysis and apply them to platformers. Part of much music analysis is finding beat patterns and determining what sorts of effects these patterns have on a listener. Using music analysis to find beat patterns in platformers may give insight into how beats should be structured to provide particular effects. For example, syncopation in music is a beat mismatch—the listener expects the next beat to be unstressed, but the composition accents it. This leads to increasing tension in the musical composition. An analogous idea for platformers might be difficulty–mixing up the next beat can throw players out of flow and make a level harder. To do this intentionally requires a more refined understanding of how beats function in platformers, which is something that music analysis may provide.

However, there is an immediate problem: Platformers may have beat (an element in a rhythm group as defined in (Smith, Cha, and Whitehead 2008)), but they don't have meter. Much of beat-based music theory assumes that a song is structured around a regular pattern of 'weak' and 'strong' beats, and this pattern is tethered to time. We refer to this regular pattern as meter. Most games don't have this temporal pacing constraint. Players can pause in the action (by using the pause button or relaxing on movement inputs, for example) and take some number of moments to size up the next set of actions to take.

This workshop paper presents some preliminary work on applying gestalt music analysis (GMA) to platformer levels. By looking at how player state changes over time, we can group similar state sequences together in a hierarchy. Player state changes based on how level obstacles are placed, tying state shift to a 'beat'. State shifts correspond to player actions, which form a 'beat' as players progress through a level. Level features, such as foes or gaps, force players to take action (and, correspondingly, change state). Even in cases where a player doesn't need to shift states (such as sprinting under a closing door when they were already sprinting), identifying that sort of mismatch can still be useful for a designer. Low level gestalt patterns show similar repetition to some common melodic lines.

Gestalt patterns, in music, work like axioms to build out more complex high-level patterns. Musical gestalt analysis highlights how the average listener perceives a piece. By clumping together the fast notes of a run into a single axiom, GMA showcases that the individual notes don't matter as much as the gesture of the run. Due to how music is written, these important abstractions can be lost. A gestalt analysis of platforms can let us describe important patterns in platformer levels and the elements that comprise these patterns.

This lets us leverage some ideas from GMA to games, allowing us to identify high points of interest or surprise in a level. Furthermore, GMA is formal enough to be automated, so we can encode the process to be performed by a level generator or game understanding AI.

Related Work

Using beats to generate new platformer levels has been well explored. (Compton and Mateas 2006) uses a context free grammar to create new platformer levels, (Smith, Whitehead, and Mateas 2011; Smith et al. 2009) uses a constraint solver to find a set of rhythm groups that fit a set of constraints, then finds a set of geometry that fits those rhythm group's constraints.

Music analysis has a different definition of beat than what is used in level generation. However, the platformer generation definition of beat ties in nicely with the concept of a gameplay gestalt. (Lindley 2002) defines a gameplay gestalt as a pattern of interaction that the player enacts or performs in order to make progress in a game. For platformers, this is similar to movement: players enact a particular set of jumps to get through a section with a lot of pits, or sprint for long distances to get through a section with a lot of falling hazards. This delimits a continuous temporal gameplay space into discrete chunks, where a player is enacting a particular gestalt. The longer a player stays in a particular gestalt, the more repetitive a segment of gameplay is (because the player is enacting the same pattern). This is a very similar definition to Tenney and Polansky (1980) view of music, where musical gestalts are segments of cohesion that are distinct from the segments before and after them. The longer a piece stays in a particular gestalt, the more repetitive it is as a single idea is being repeated and stretched over more time.

This work adapts Tenney and Polansky's model to work with platformer play traces, and hopefully in the future, will use a similar model to eventually be able to generate new levels.

Gestalt Music Analysis

Gestalt analysis got its start in perceptual psychology (Wolters and Koffka 1936) before being quickly adapted to musical analysis, (Ehrenfels 1937). Tenney and Polansky encoded a gestalt music analysis in a program, and got useful results looking at monophonic flute music. However, most modern GMA methods are not conducive to being automated– for example, generative tonal harmony combines Schenkerian analysis, gestalts and generative grammars (Lerdahl and Jackendoff 1987), but is not formal enough to be encoded as a program (it leaves several important decisions up to the theoretician to justify). A first step for using these methods for games is to add enough formal rigor to allow for automation.

The defining idea of GMA is to break music up into temporal units called gestalts. Gestalts are complete over the entire piece of music (all notes of a piece of music belong to at least one gestalt) and hierarchical (higher level gestalts are composed of lower level ones). Each gestalt is a single, internally consistent sound pattern, which is segregated from the gestalts that precede and follow it.

In order to break apart a piece of music, Tenney and Polansky (Tenney and Polansky 1980) looked at several qualities of each note. In figure 1, the qualities are duration (how many eighth notes a single note takes up in time) and pitch class (musical abstraction of frequency of the sound wave for that note). A new gestalt boundary is drawn between two notes i and $i+1$ if the distance between them is greater than the distance between notes $i-1$ and i and notes $i+1$ and $i+2$. Distance, for this model, is a weighted linear combination of the euclidean pitch and time distance between two notes (the reference figure uses a weight of 1 on the parameters, but more complex models are discussed).

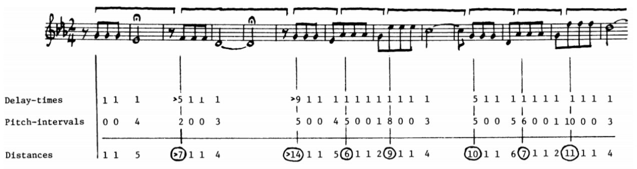
Delay-times 1 1 1 >5 1 1 1 >9 1 1 1 1 1 1 1 1 1 1 1 5 1 1 1 1 1 1 1 1 1 1 1
Pitch-intervals 0 0 4 2 0 0 3 5 0 0 4 5 0 0 1 8 0 0 3 5 0 0 5 6 0 0 1 10 0 0 3
Distances 1 1 5 >7 1 1 4 >14 1 1 5 6 1 1 2 9 1 1 4 10 1 1 6 7 1 1 2 11 1 1 4

Figure 1: A Gestalt Analysis from (Tenney and Polansky 1980). The melodic line from Beethoven's 5th symphony can be broken into discrete segments based on the pitch and duration of the previous and next notes. These segments correspond to what a listener hears, and each boundary identifies a point of interest in the piece of music.

(Lerdahl and Jackendoff 1987) and other applied versions of GMA discuss that the perceptual level of gestalts isn't this low level note grouping, but the level above this one (the first group of groups). For the average listener, music is not perceived as groups of notes, but as groups of groups of notes. However, this first level of grouping gives us the building blocks for the next ones.

Application To Games

Each low level gestalt, for music, represents a single musical idea (the repeated notes and final jump in figure 1). Gestalts in games are a little different. Lindley (2002), in talking about ludo-narrative dissonance, defines a 'gameplay gestalt' as a pattern of interaction that brings success or progress in the game. Players learn the rules, and from their understanding of those rules, perform a gameplay gestalt that optimizes for their own goals within the game. This means that these gestalts change over time, as player goals change, the game changes how to achieve those goals or players develop a deeper understanding of the rules.

In theory, then, if we can chart out gameplay gestalts, we can find points when players chose to perform a new gestalt. This may indicate that the underlying level changed and required a new pattern of interaction in order for a player to progress, or that a player's goals changed within the game and they adopted a new input paradigm.

Gameplay and monophonic melodic lines are very different, but both can be considered as a set of state changes over time. These states both have the same temporal relations (if action/note A happens before action/note B in the trace/music, then they also have this same sort of temporal relation when played/performed). Gestalt analysis already takes this 'states through time' view of music, so it seems applicable to games.

We looked at the Platformer Experience Dataset (PED) to try and map gestalt analysis onto games. PED (Karpouzis et al. 2015) is a dataset of gameplay traces from *Infinite Mario*

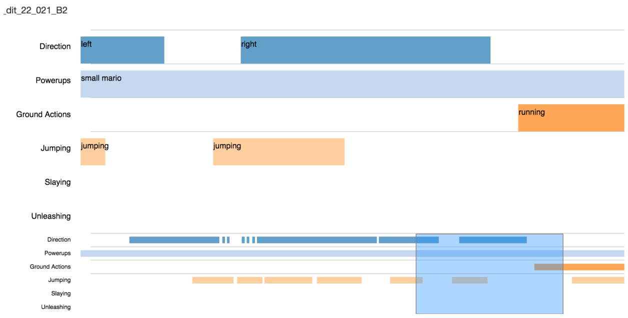

Figure 2: Mario states from a PED trace. Each track corresponds to a different set of non-overlapping potential states. Filled in blocks are when that state is true. The bottom part of the figure shows the full trace, the upper part is a zoomed in subsection. Overlap is where Mario is in two states at once. Overlapping states are concatenated in the remaining figures.

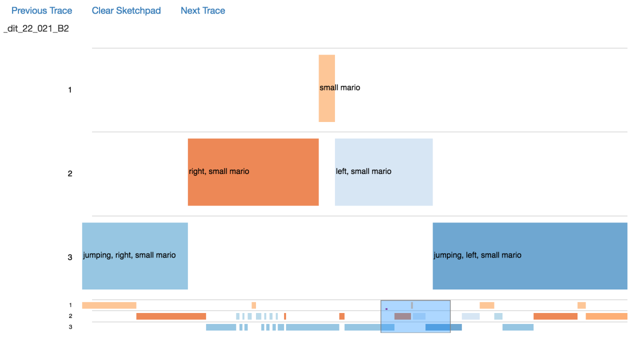

Figure 3: Mario states from a PED trace. Each color corresponds to a particular state. States on higher numbered lanes have more simultaneous sub-states than states on lower numbered lanes. Lower part is the full trace, the upper part is a zoomed in subsection.

Bros along with visual recordings of players playing. We only look at the gameplay trace side of the dataset, although future work could be done on correlating visual cues of surprise with gestalt boundaries.

PED also does not contain level information in it's gameplay traces. The lack of context in state actions makes it hard to find context for Mario's actions. Gestalt analysis is less useful without this contextual information, but can still reveal interaction patterns.

For PED, we can chart Mario's state as he goes across the level. We're primarily interested in input state, but we also chart Mario's powerup state as well. By using a linear model similar to the one used for music, we can see if gameplay traces have similar boundaries and patterns. PED tracks Mario's state individually, as a set of onset and ending times for various states Mario can be in. Several of these states can overlap (such as powerup state, run state and movement direction), so we concatenated these combined states into a single super-state. The non-concatenated version is show in in 2, the concatenated(without gestalt boundaries being drawn) is shown in figure 3.

Gestalt Boundary Model

For gameplay traces, we look at state distance by taking a weighted linear combination of the individual parameter distances. 4 out of 5 parameters directly correspond to player input. Powerup state is significant because it changes how much risk a player can take (big Mario can take a hit before a game over). The five state parameters are:

1. Movement Direction (left, right or none). This parameter captures which direction Mario is moving in, if he's moving at all. Going from left or right to no movement has a distance of 1, switching direction has a distance of 2.
2. Powerup State (small Mario, big Mario, fire Mario). This parameter captures if Mario has any power ups. Going from small to big, big to small, big to fire, or fire to big has a distance of 1. Any other transitions have a distance of 2.
3. Ground State (crouching, running, none). This parameter captures if Mario is running, crouching, or just standing. Going from crouching or running to none, or starting a crouch or a run, has a distance of 1. Any other state transition has a distance of 2.
4. Airborne State (jumping, none). This parameter tracks if Mario is airborne or not. Switching between these states has a distance of 1.
5. Time. This parameter tracks the onset time between two states. The distance is just taking the difference of state onset times.

The distance metric used is the euclidean distance, so the full state distance calculation is below, where s is a state and n is the number of parameters that make up a state (in our case, 5):

$$dist(s^1, s^2) = \sqrt{\sum_{i}^{n}(paramWeight_i * paramDist_i(s_i^1, s_i^2))^2}$$

To start, we did not weight any state parameters. This put a large weight on time, as it is the only parameter that can range beyond 2. This occasionally lead to some strange boundaries being drawn, as the model is overly sensitive to state duration, as shown in figure . Early in the trace (the blue section), a boundary is drawn between a small flurry of jumps and run actions. This is likely actually all part of a single gestalt; the player is very briefly dealing with a dense enemy pattern or set of complicated set of geometry. However, even with this large sensitivity, low level structure does start to reveal itself—boundaries are correctly drawn towards the end of the trace, where Mario is repeating a jump action. This is similar to the gestalts shown in figure 1, repetition of this jump action works similarly to the repetition of the three eight notes and single half note in the melody of Beethoven's 5th.

Experimenting with tunings for the distance weights leads to (unsurprisingly) different boundaries. Tweaking the

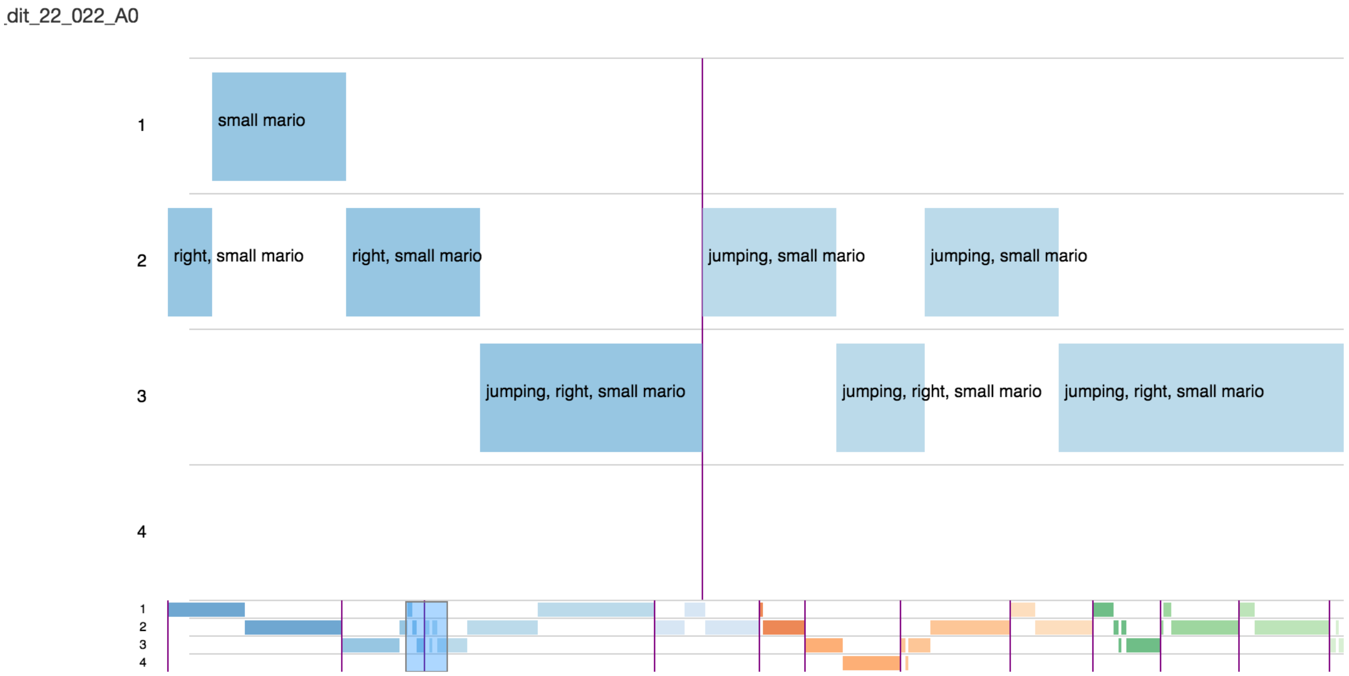

Figure 4: Low level gestalt boundaries. Purple lines and colors show each found gestalt. Compare and contrast with figure , which shows a different set of gestalt boundaries for this trace

weights, where time is given very little weight (0.01), direction, ground state and airborne state are all given moderate weights (0.75, 0.25 and 0.5 respectively) and most of the weight is on Mario getting or losing a powerup (1); we start to get traces that seem to graph important shifts. The flurry of actions shown in figure no longer have a boundry between them, grouping them as one unit—much like how a flurry of notes would be grouped as one unit. Furthermore, we still have the useful repetition at the end of the trace. However, the boundaries still don't quite line up, as they seem too conservative at the beginning of the trace. The blue flurry of state changes should be in its own gestalt, and not tethered to the relatively long pause where the player is stationary as small Mario.

Future Work

We can trace low-level gestalts in gameplay traces that seem analogous to gestalts in music. However, we don't know if the gestalt boundaries are consistent with themselves: if this is graphing an important concept, than for multiple traces of the same level, the gestalt boundaries should be similar (as player actions are constrained from level geometry). We can use this internal consistency metric to tune the model.

The presented model only finds the first level of gestalts. The real goal would be to find the next level of gestalts (the first grouping of groupings). It's this level that's commonly used in musical analysis, and these abstract levels that correspond to human perception (Lerdahl and Jackendoff 1987). We can label each of these low level gestalts with the dominant state, or blend states for a single gestalt label with an weighted averaging metric similar to (Tenney and Polansky 1980). Looking at the next level boundaries from these states may grant insight into when and why players adopt new strategies, and may lead to a deeper understanding of when games are surprising or at least interesting. It may also give an algorithmic way to find high level state boundaries in games—when a MOBA shifts from the early game to the mid-game is fuzzy, and high level gestalts may shed light on these sorts of gameplay shifts.

The presented model has only be used with PED. PED is a dataset of playtraces on generated *Infinite Mario Bros* levels. Before doing serious analysis on generated levels, we'd like to have a library of patterns that show up from a gestalt analysis of known well-authored levels.

Another weakness in using PED data is that it only captures player state. The model can not reason about what caused a player to jump, only that a player has jumped. Adapting the model to work with playtraces that also contain level geometry data can lead to gestalt boundaries that can reflect both a level change as well as a player goal change.

Along those lines, we tried to keep the model as domain agnostic as possible. However, some specific knowledge is required in order to come up with what parameters compose a particular state, and distance functions on those parameters. Going through with this same sort analysis for other games can give us a sense of how general this model is. We'd expect adaptation to be highly difficult for walking simulators, as most of the gestalt changes a player might go

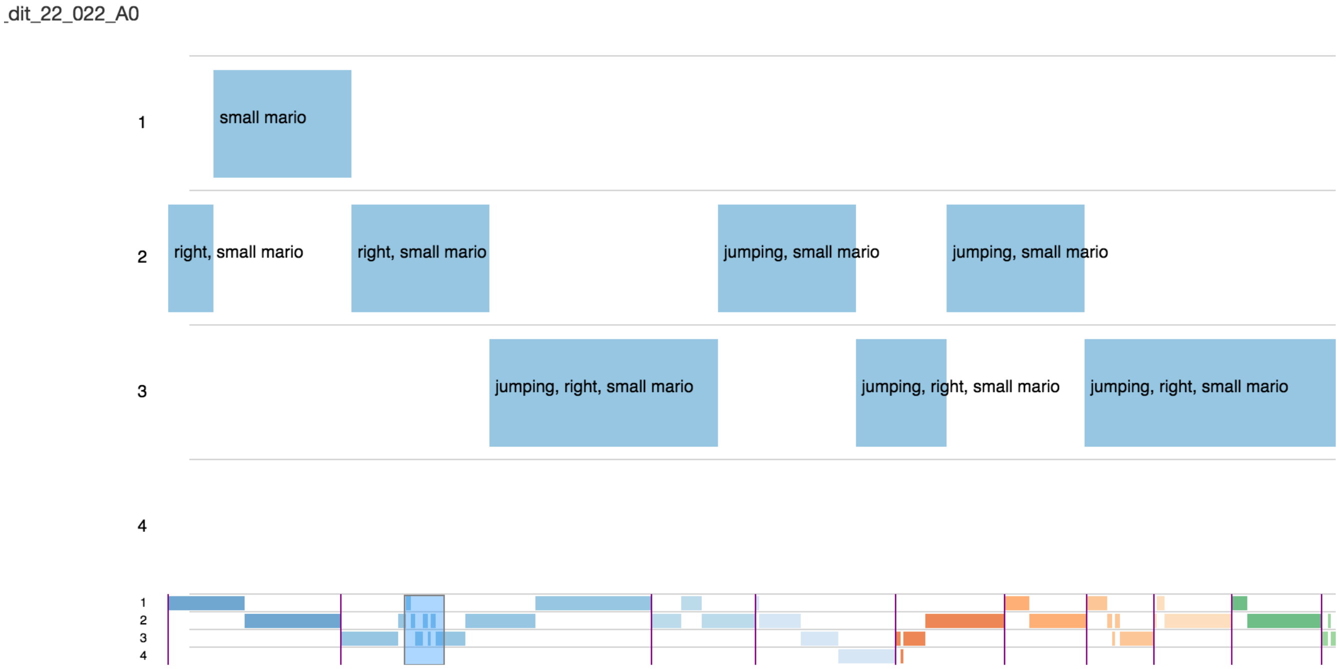

Figure 5: Gestalt graph with weights. We start to see what looks like important repetition and a solid development of patterns in the trace.

through are not reflected in their inputs. A player may find a delightful hut in the woods that changes how they view the game (which would be, according to the theory, an important gestalt shift) but that change may never get reflected in how they move about the space.

Finally, we'd like to, eventually, hook up this model to a generator. A well formed understanding of gameplay gestalts and how they could relate to level geometry could let a level generator try to generate a level that has a particular gestalt pattern. We can never be truly sure that a player will enact gestalts exactly how we'd like them too, but we can try to design around players needing to adopt new gestalts at particular times.

In conclusion, we believe that gestalt analysis is starting to show a interesting way to look at games. Potentially, techniques from music theory could be adapted to be a general way to segment out play traces, describe particular properties of levels, and work as a proxy for designer intent in a gestalt-aware level generator.

References

Compton, K., and Mateas, M. 2006. Procedural level design for platform games. In *AIIDE*, 109–111.

Csikszentmihalyi, M. 1990. *Flow: the psychology of optimal experience*. Harper & Row.

Ehrenfels, C. v. 1937. On gestalt-qualities. *Psychological Review* 44(6):521.

Karpouzis, K.; Yannakakis, G. N.; Shaker, N.; and Asteriadis, S. 2015. The platformer experience dataset. In *Affective Computing and Intelligent Interaction (ACII), 2015 International Conference on*, 712–718. IEEE.

Lerdahl, F., and Jackendoff, R. 1987. A generative theory of tonal music.

Lindley, C. A. 2002. The gameplay gestalt, narrative, and interactive storytelling. In *CGDC Conf.* Citeseer.

Smith, G.; Treanor, M.; Whitehead, J.; and Mateas, M. 2009. Rhythm-based level generation for 2d platformers. In *Proceedings of the 4th International Conference on Foundations of Digital Games*, 175–182. ACM.

Smith, G.; Cha, M.; and Whitehead, J. 2008. A framework for analysis of 2d platformer levels. In *Proceedings of the 2008 ACM SIGGRAPH symposium on Video games*, 75–80. ACM.

Smith, G.; Whitehead, J.; and Mateas, M. 2011. Tanagra: Reactive planning and constraint solving for mixed-initiative level design. *IEEE Transactions on Computational Intelligence and AI in Games* 3(3):201–215.

Tenney, J., and Polansky, L. 1980. Temporal gestalt perception in music. *Journal of Music Theory* 24(2):205–241.

Wolters, A., and Koffka, K. 1936. Principles of gestalt psychology.