Lecture8 Response Metrics

1. Response Mapping

Response Update

response, three essential steps

- input signal comes in
- input signal is interpreted and filtered
- input signal modulates some parameter in the game

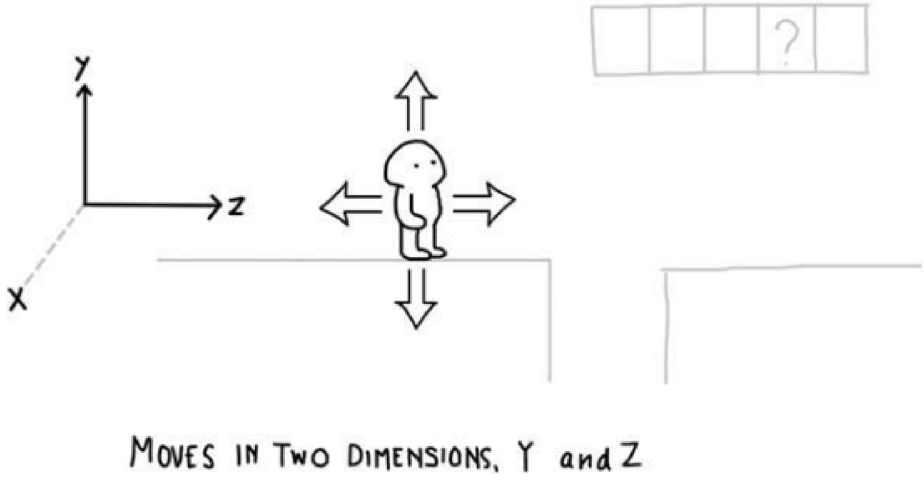
Input Mapping

a input can be mapped as

- an avatar (rotation and position change)
- the creation of a new entity
- the playback of a linear animation
- a change in one or more parameters in a simulation

Mapping to Motion

the input signals will be mapped to the motion of the avatar

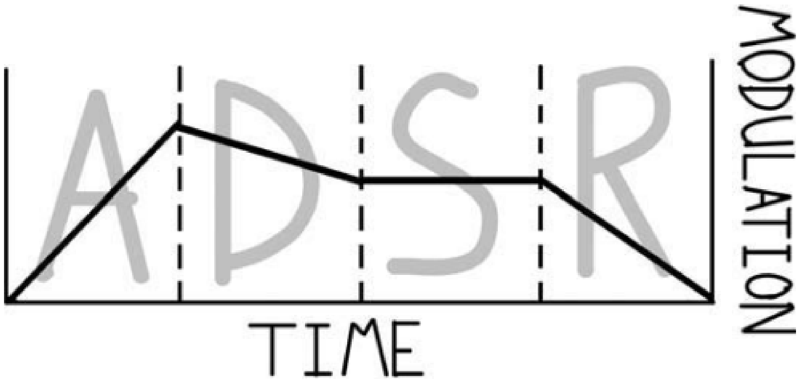


- type of motion: linear vs. rotation
- dimensions of motion: in what dimensions, X, Y, or Z, does the avatar move or rotate
- absolute or relative motion: what frame of reference does the motion use
  - absolute: avatar movement related to the world
  - relative: camera
- position vs. rate/magnitude
  - position: mouse cursor is mapped to changes in position

  - rate/magnitude: pushing the thumbstick is mapped the rate at which the avatar turns
- direct vs. indirect control
  - direct: modify the character directly
  - indirect: add forces to a simulation or cause another object to move or rotate (Zuma is mapped to the direction of the frog will face)
- integrated vs. separate dimensions: does the input change one parameter in the game or many

2. Attack, Decay, Sustain and Release

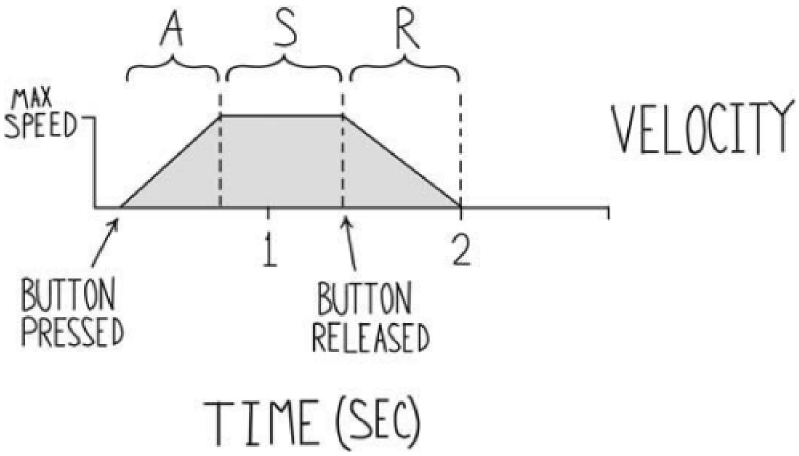
ADSR Envelope



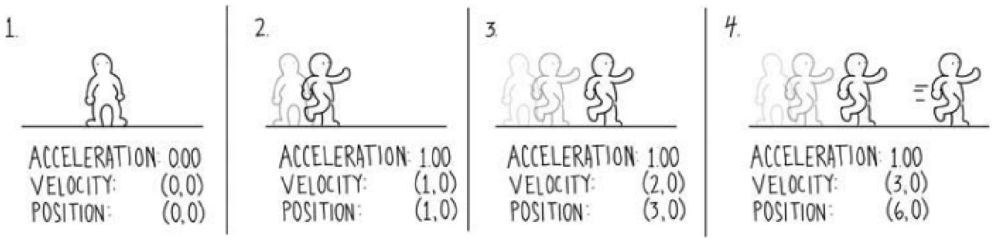
- the modulation of a parameter over time will have some kind of curve
  - A: Attack
  - D: Decay
  - S: Sustain
  - R: Release
- such envelopes are used to describe the modulation of the sound of musical instruments
- ADSR envelopes are often used to modulate the output of **digital instruments** to make them sound like their physical, real-world counterparts

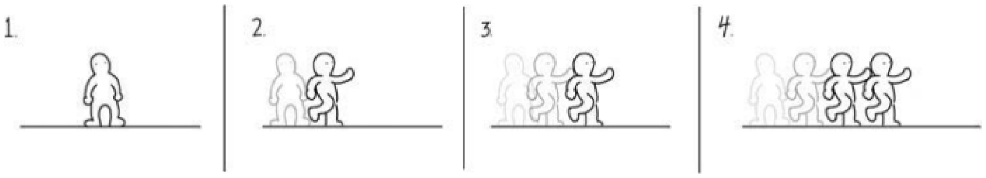
Simulation

Mario ADSR

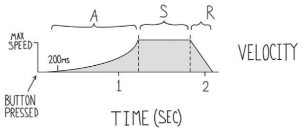
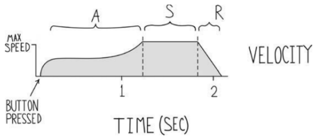


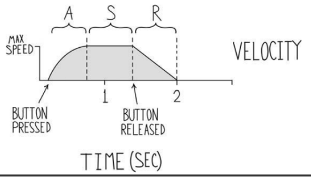
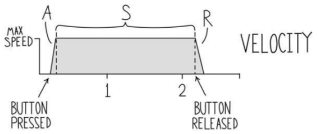
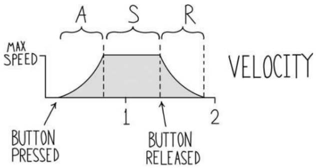
- vertical axis: movement (speed)
- ADSR
  - A: Mario ramps up to his maximum speed
  - ~~D: no decay~~
  - S: sustain as long as the button is held
  - D: a long release when the button is released





Analysis

Graph	Description
	A longer Attack phase results in a floaty of loose feel. If it is too long, it seems to be no immediate response to the input. This is problematic because it starts to erode the impression of instantaneous response. There may be some small change happening immediately, but if the player can't perceive it, the game feels **unresponsive**
	An envelope like this will make player feel both **instantaneous-feeling response and a loose, organic feel**

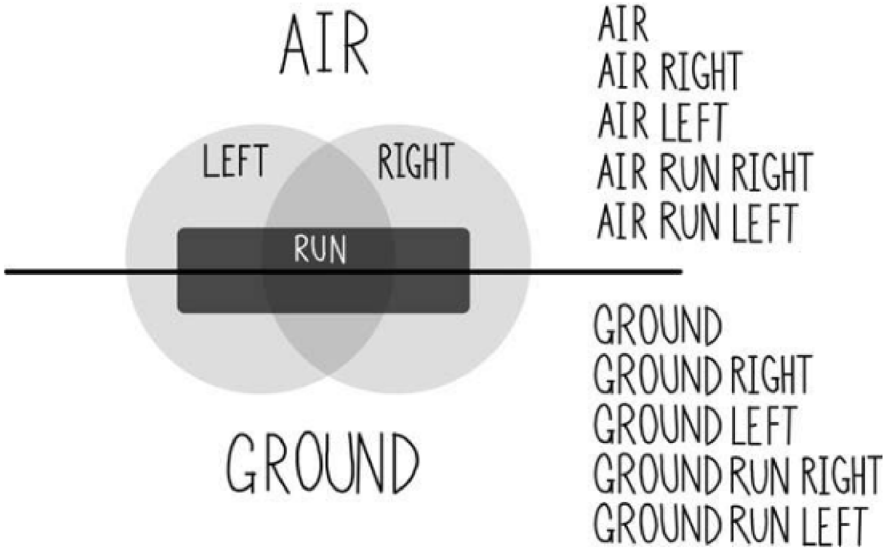
Graph	Description
	A short attack phase will tend to feel **tight and responsive**
	When the attack phase is short and when there is more linear progression from off to on, player feel as **twitchy**. The attack is totally linear and very short, the controls can feel stiff.
	This speeds up gradually, giving player a loose, organic feel.

- All these sensation: **floaty, twitchy, tight, loose, unresponsive** —all exist on the same continuum
- They're just slightly different envelopes, slightly different modulations of motion over time
- That motion could be direct or indirect, a force or a rotation; regardless, changes in attack will alter the feel of control

State Changes

- Simulation is state changes
- States are artificially constructed changes in circumstance that modify the meaning of incoming signals
- There is greater expressivity when **inputs are mapped to different responses** across states which are altered and maintained by the simulation itself.

Mario Example

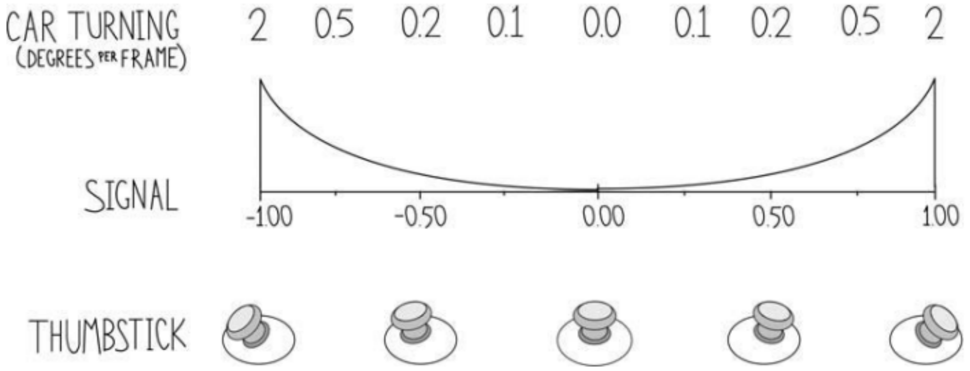


- Income signal: left, right and jump

- Mario has a `ground` state and an `air` state
- Mario's potential for movement—his physical properties—change when he's on the ground or in the air
  - ground state: left and right buttons map to certain additive force
  - air state: the strength of his left and right movement is greatly reduced

Filtering

- Input signals come in form the input device in various forms
  - boolean
  - float
- It is possible to map "raw " input either directly to a response or to a force or other modification of a simulation
- Most input signals are **modified** before being mapped to a response
- It is also possible to have **complex, non linear transformations** applied to input signals as they come in



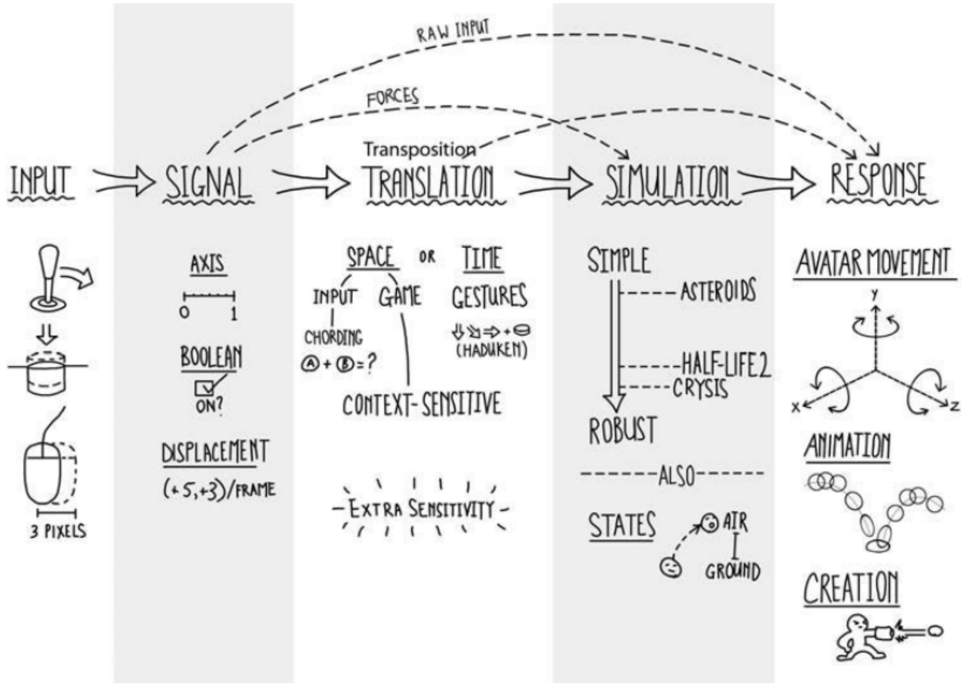
- GTA4 featuring a driving metaphor, instead of a constant car steering ratio (1 degree of steering wheel turn 2 degrees of car turn) the amount of turning changes across the input space
- There is the possibility not only for modifying input signals before passing them on, but for **creating entirely new signals** by further interpretation of the incoming signals
  - In some game, there requires a certain sequence of input over time to trigger a specific attack
- Another way in which it's possible to create additional sensitivity through interpretation is **spatially**, either across game space or input space
  - Press A-button may have one meaning, and pressing B-button may have a different meaning, pressing both simultaneously yields a third response

    It assigns a different meaning to a combination of inputs than it does to each of those inputs individually
- **Context sensitivity**: the position of the character in the game world can alter the meaning of a particular input.

Relationships

Individual mechanics—mappings of one input to one response— **work in concert** to produce an overall feeling of control.

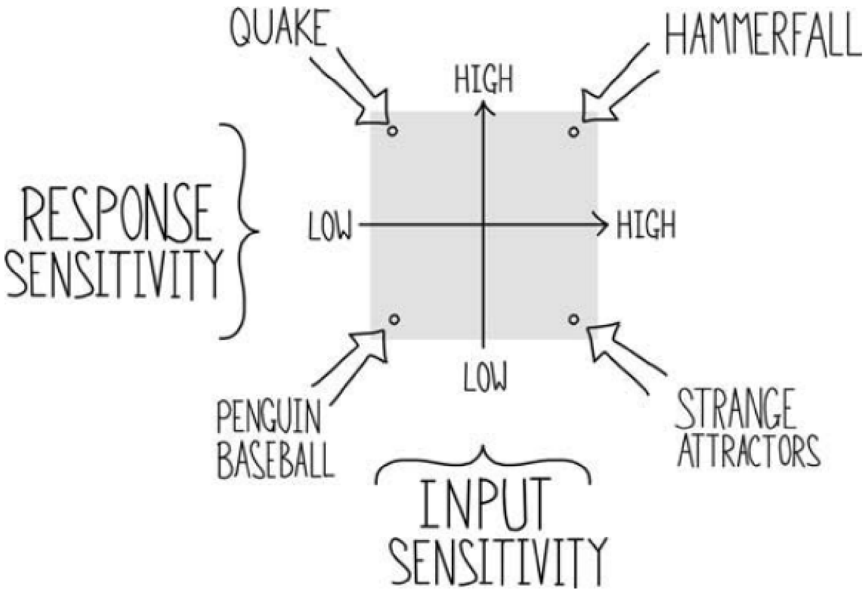
From Input to Response



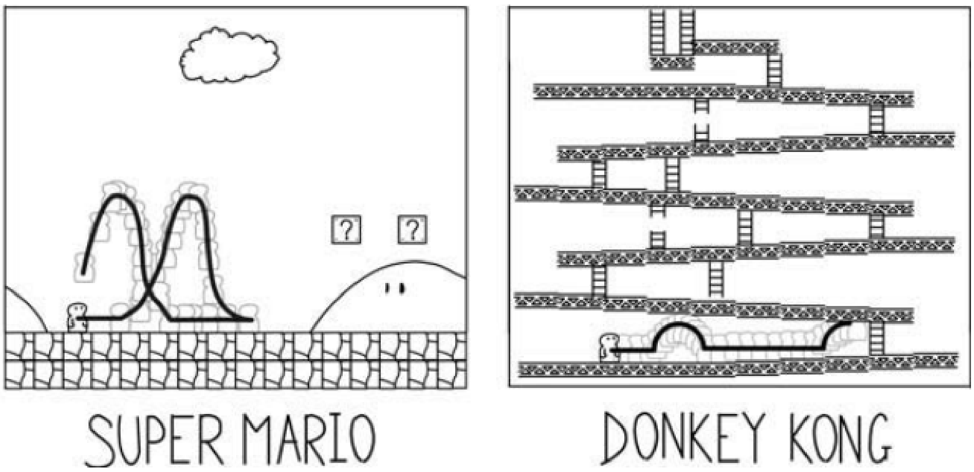
- From the physical manipulations of various inputs, the input device generates and sends to the game corresponding signals
- A raw input can be
  - mapped directly to response (mouse cursor)
  - feed directly into a simulation
- Or: Some kind of filtering happens, where the input signal coming in is altered in some ways before being passed along to simulation or response
- Simulation layer represents the game's internal model of reality, the one which the player interacts with via input
- Game response to the signals it receives from transpose, raw or simulation

3. Input and Response Sensitivity

Virtual Sensitivity



- This is a **soft metric**, of course, but useful for comparing the expressivity of two different games.



- Super Mario Brothers: a very low-sensitivity input device but has a very sensitive response
- Donkey Kong: maps a relatively high-sensitivity input device

4. Summary

Metrics

- Hard
  - How many objects the player controls
  - The dimensions, type and frame of reference for the movement of each avatar
  - The ADSR envelope representing each modulation of a game parameter by an input over time
- Soft
  - The overall sensitivity of the system as a function of its input and response sensitivity