GAT 261

User Experience Design II

Instructor

- Rich Rowan
- rrowan@digipen.edu
- Cell: 206-898-2955 (texts welcome)

Office Hours

- Tues/Weds 2pm-4:30pm

DigiPen INSTITUTE OF TECHNOLOGY

Game Feel Metrics II & HUDs

- ❑ Response Metrics
- ❑ Control Envelopes

PLEASE SILENCE
ALL ELECTRONIC DEVICES

THANK YOU

RESPONSE METRICS

Measuring Impact of Response on Game Feel

What is Response?

Response is the processing and output of a system to input from the player. Processing means the input is interpreted and filtered to map to:

- Change in the position or orientation of an avatar
- Creation of a new entity (e.g. a bullet)
- Linear playback of an animation (e.g. fighting game "move")
- Change in a parameter in the game simulation (gravity, friction, animation playback speed, etc.)
- Change in interface state (e.g. opening a menu)

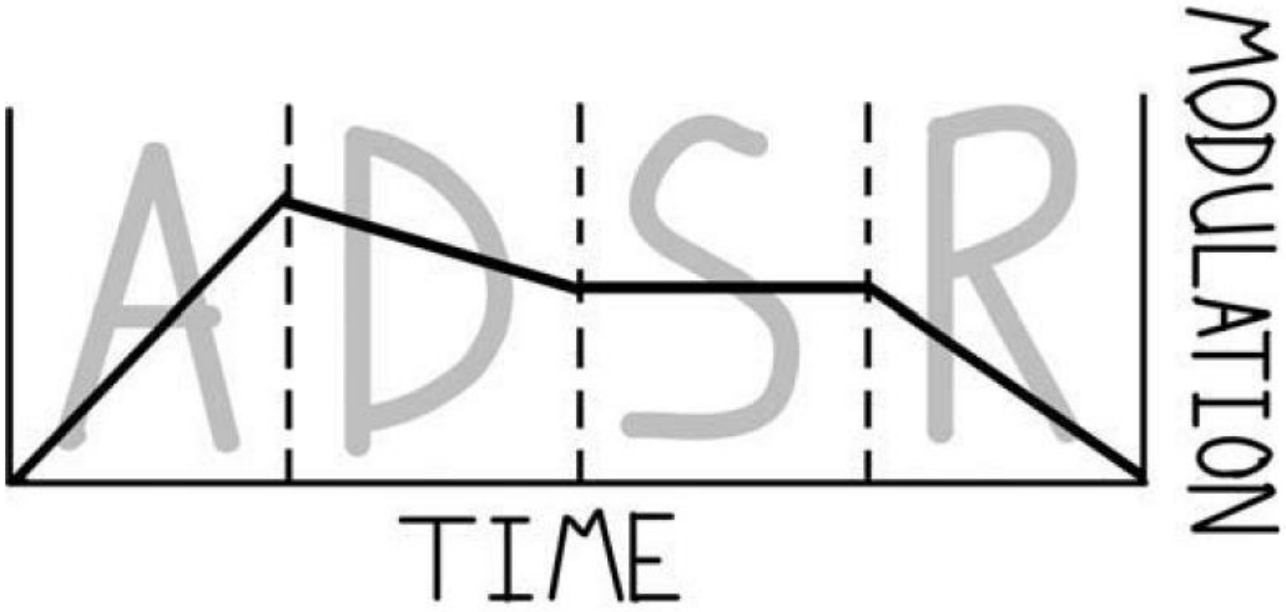
Avatar Response – Motion

When mapping an input to avatar response to create motion, the response can be measured with:

- Type of Motion
  - Linear vs. rotation (does the avatar move linearly or rotate?)
- Dimensions of Motion
  - In what dimensions—X, Y, and/or Z—does the avatar move or rotate?
- Frame of Reference
  - Absolute vs. relative (is the motion relative to the avatar, camera, or some other point in the world?)
- Position vs. Rate/Magnitude
  - Mouse cursor changes position, whereas accelerator in racing game changes rate or Halo changes turn speed
- Control Type
  - Direct vs. indirect (does the input modify the avatar directly or change forces in a simulation which affects the avatar? Zuma is a good example of direct control type.)
- Integrated vs Separate Dimensions
  - Does the input change one parameter or many? Geometry Wars maps both thrust and rotation of the ship to the left thumbstick, for example.

ADSR Envelopes

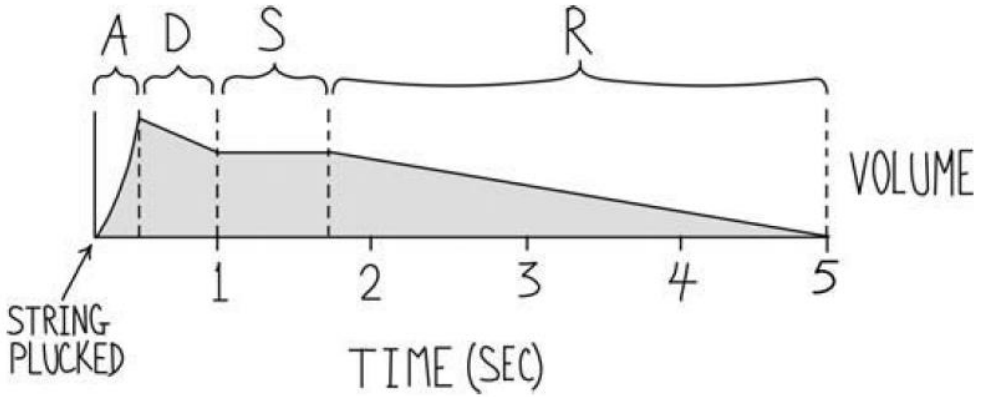
- Regardless of what parameter an input is mapped to, the modulation of a parameter over time will have some kind of curve. ADSR envelopes are a way of describing this.
- ADSR stands for Attack, Decay, Sustain, Release.



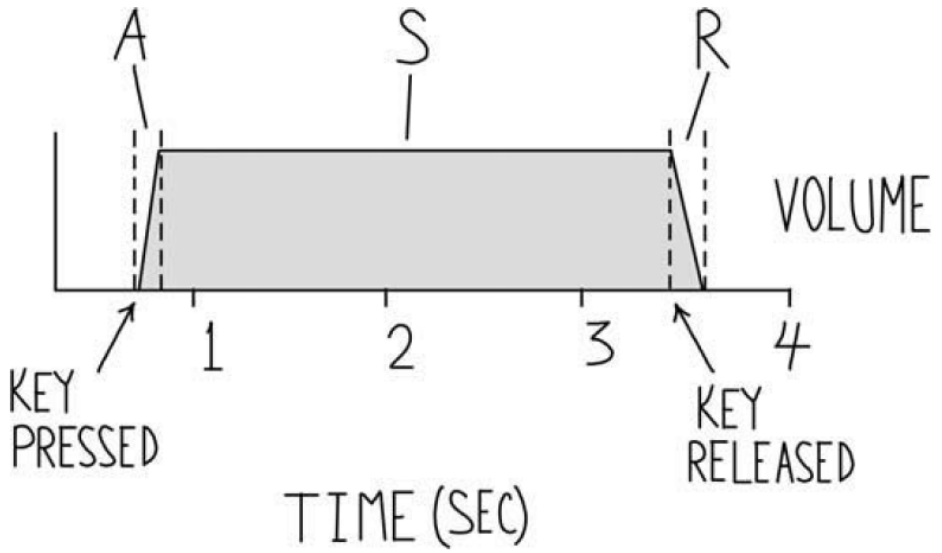
ADSR Envelopes - Sound

This terminology comes from sound:

- Attack is the time from the initiation to the highest magnitude (loudness)
- Decay is the time from the highest magnitude to a stasis point
- Sustain is the time it maintains at the stasis point before starting to fall off naturally or due to cease of input.
- Release is the time from the start of the fall off to zero magnitude (silence)



The ADSR envelope for a guitar's volume when a string is plucked.

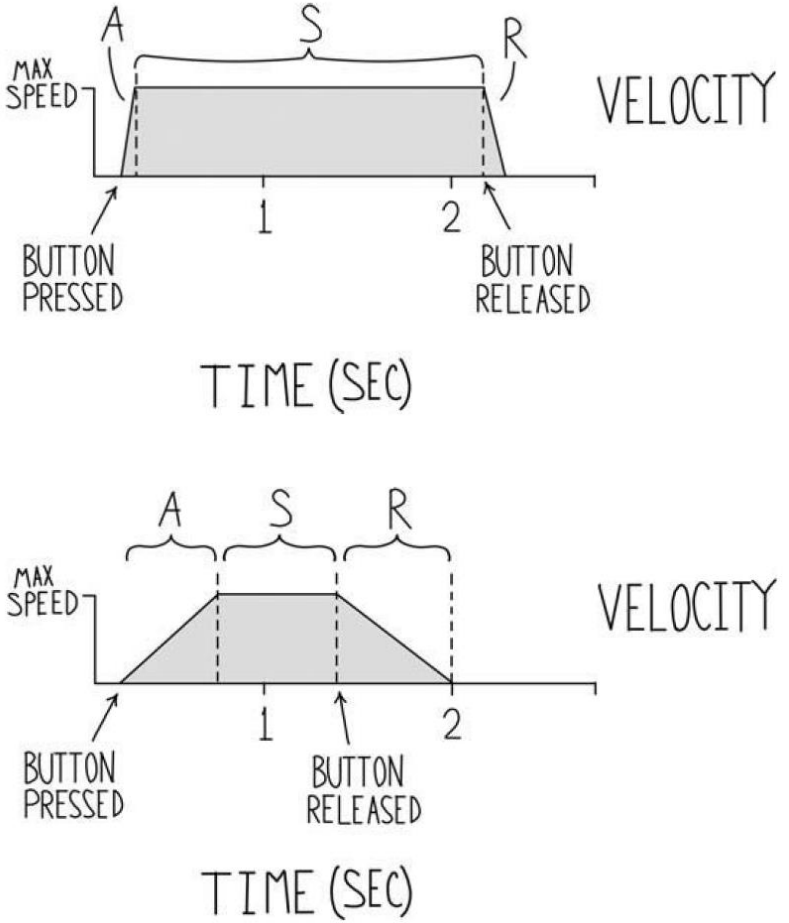


The envelope for a pipe organ is much more rigid.

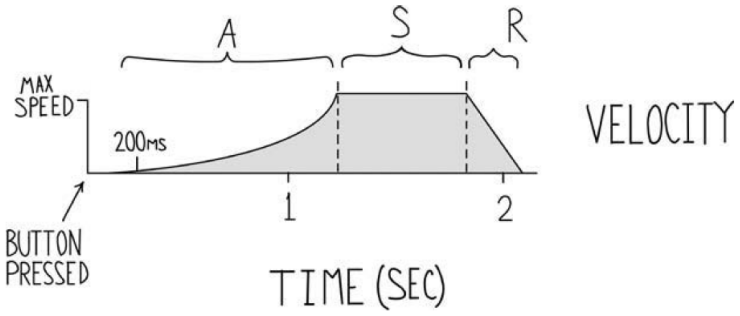
ADSR Envelope – Motion

Let's consider the change in how Mario moves horizontally:

- Donkey Kong
  - No attack (other than processing time)
  - No decay
  - Sustain as long as the button is held (max speed)
  - No release (other than processing time)
- Super Mario Brothers
  - There is an initial attack phase as Mario ramps up to his maximum speed.
  - No decay state.
  - Sustain as long as the button is held.
  - Release is the mirror of the attack change when the button is released.

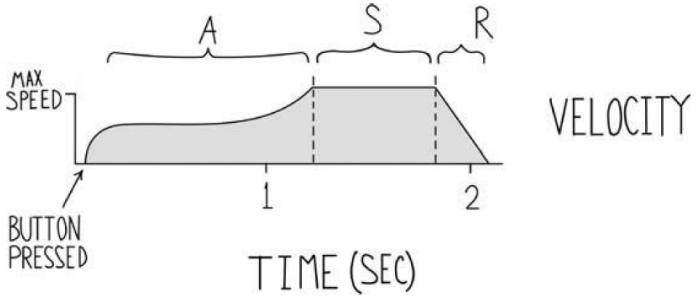


ADSR Motion Envelopes Compared



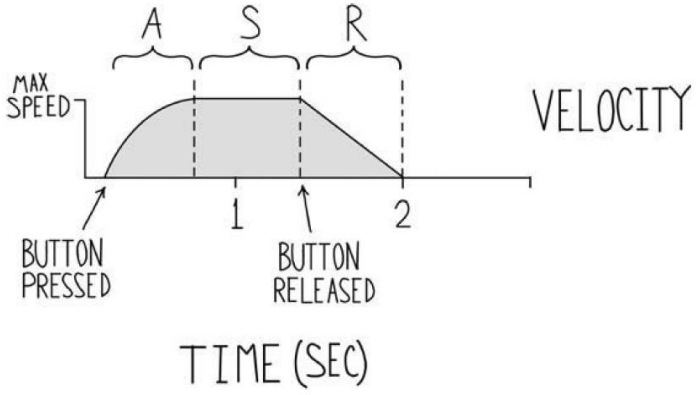
Unresponsive

(more than one 100ms perceptual processing cycle)



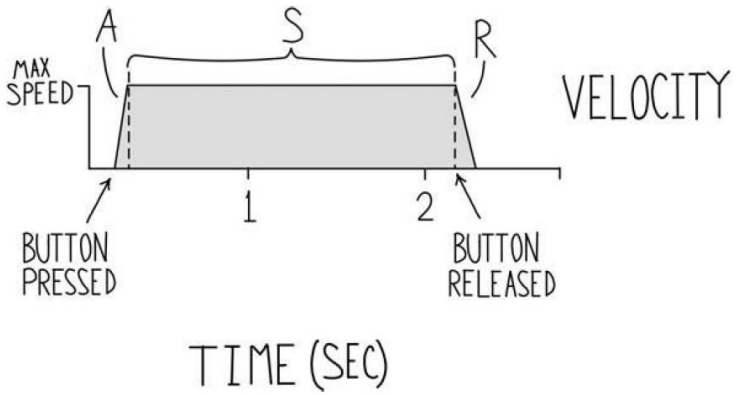
Loose (Floaty)

(strong initial attack, but slow build)



Tight (Responsive & Organic)

(short and curved attack and release – various flavors)



Rigid (Twitchy)

(extremely short, linear attack and release)

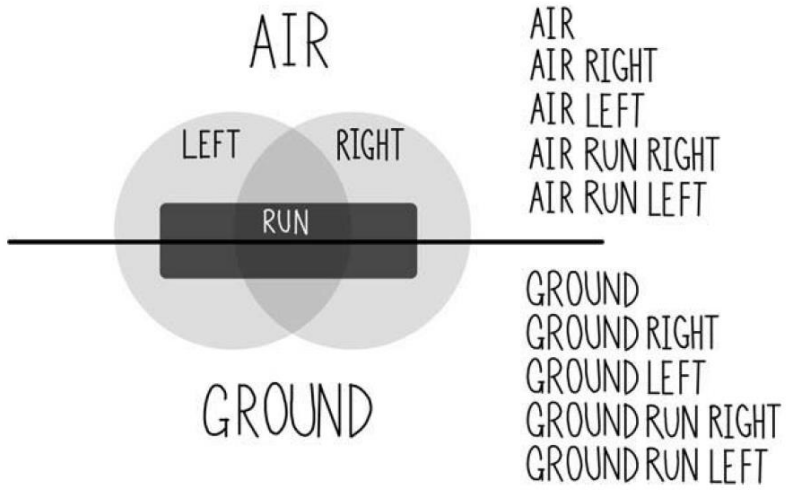
Beware of Decay

- When decay exists in games, it is usually by accident.
- This is almost never desirable, as it results in rapid button mashing which causes physical fatigue.
- Consider early beta versions of Counter-Strike which allowed angular skating by pressing forward and left/right in rapid alternation. This allowed the speed to exceed the maximum sustain which created an exploit.

Simulation State Changes

An interesting, measurable feature of a simulation is a change in state. Simulation states are changes in circumstance that modify the meaning of input signals.

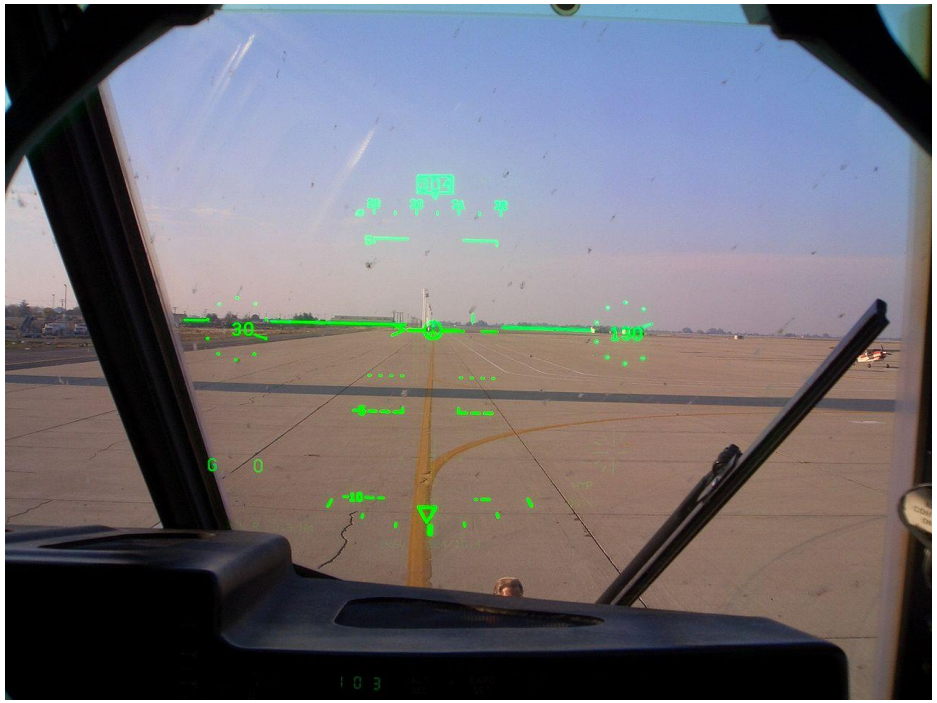
- Mario, for example, ostensibly has three inputs for movement – left, right, and jump.
- Mario normally has a “ground” state and an “air” state (and in some games, a “wall” state) that modifies the meaning of inputs. Certain power-ups can alter this state, as well, such as the Tanooki suit.
- While in the air state, the strength of the left/right input is reduced and the jump input is disabled. This allows for greater expression with the same controls.



HUD Design

Traditional HUD Basics

- HUDs, or head-up displays, originated with aircraft instrumentation displays.
- HUDs have been around since the very earliest days of video gaming, inherited from pinball backboards.
- The best HUD is no HUD at all. Unless you need one…



"C-130J Co Pilot's Head-up display" by Telstar Logistics - flickr. Licensed under CC BY 2.0 via Commons - https://commons.wikimedia.org/wiki/File:C-130J_Co_Pilot%27s_Head-up_display.jpg#/media/File:C-130J_Co_Pilot%27s_Head-up_display.jpg

Do You Really Want to HUD Me?

- Do you really need a HUD?
  - Information Architecture
    - What decisions need instant access to this information at all times? Probably a lot less than you think. Don't dress your displays the same way it's always been done unless there's a good reason.
  - Diegetic Displays
    - Can you push this information into the game world?
  - Audio Cues
    - Can you use an audio cue for this information?
  - Contextual Displays
    - Don't be afraid to have HUD elements come and go as they're needed.

HUD Theme

- The more the HUD is themed to the game world, the less intrusive it will appear. Can you identify the theme of the game just from the HUD?
- This is more than just visual theme. Ask yourself: “Based on the world we've designed, what is the best way to convey this necessary information to the player?”

http://www.gamasutra.com/view/feature/130948/off_with_their_huds_rethinking_.php

General HUD Advice

- Reduce/remove elements if possible
- Contextually hide HUD elements if they’re not always important
- The more important the message, the less frequent and the more visual impact it should have
- Make use of peripheral vision in your HUD – draw the player's attention when you need to
- Animate the HUD to draw attention to it

HUD Colors

- Picking colors for HUDs is really hard, because the HUD has to work over every background.
- If your art director hasn't made the background relatively consistent in saturation, value, etc. this becomes extremely difficult and you may have to violate your color language.
- More than any other feature of your user interface, making sure that it works with color deficient vision is essential.

Redundancy of Information

- In a first person shooter, there is often an indicator around the reticle for damage, but half the players may not notice.
- Doing it on the periphery often works for the other half of the audience, but then the half that would notice the reticle may not notice this.
- Phantom Army, a first-person shooter by Zombie, ended up doing both.
- Redundant info CAN be okay. Focus redundancy on information you are pushing to the player, not that the player is pulling.

Dynamic HUD Transitions

When you have HUD elements contextually appear/disappear from the screen, consciously decide HOW they appear (don't just pop in). A few techniques:

- Fade
  - Fading into view. Most appropriate for technology or possibly magic.
- Slide In/Out
  - Slide from/to off-screen. Consider what that area of the screen "means" in your game.
- Grow/Shrink
  - Often you may want a HUD element to "pop out" of an on-screen element when a context is activated.
- Animate
  - If you have a diegetic UI, this may "fold out" of an existing element on screen (e.g. a wrist watch with a pop-up screen).

HUD Updates

When making an update to your HUD, you should usually have all four of these sub-steps.

- Attract
  - Animation/sound to draw the player's attention
- Signal
  - Foreshadow what you're going to update (e.g., yellow on a health bar, glowing numbers on a digital display, etc.)
- Update
  - Animate the update and possibly add sound to show what is changing
- Resolve
  - Animation/transition to return to the default/new state.


See You Next Class

Links

- http://blog.digitaltutors.com/designing-a-hud-that-works-for-your-game/
- http://www.hudsandguis.com/
- http://gamesinspiration.com/tags/hud
- http://gamedevelopment.tutsplus.com/tutorials/game-ui-by-example-a-crash-course-in-the-good-and-the-bad--gamedev-3943
- http://www.thewanderlust.net/2010/03/29/user-interface-design-in-video-games/