

Game Experience Design

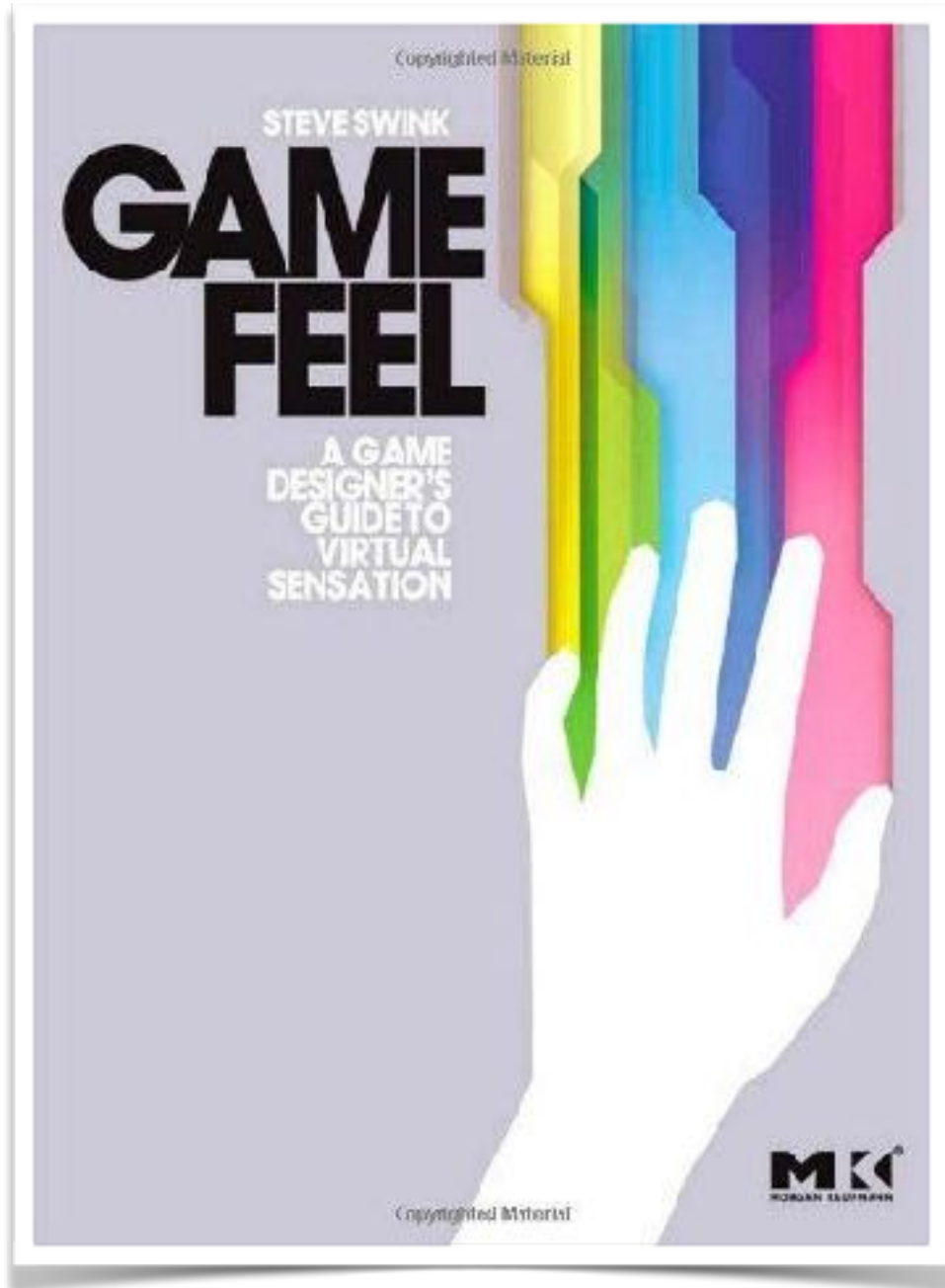
10 Game Feel

Prof. Dr. Jochen Koubek

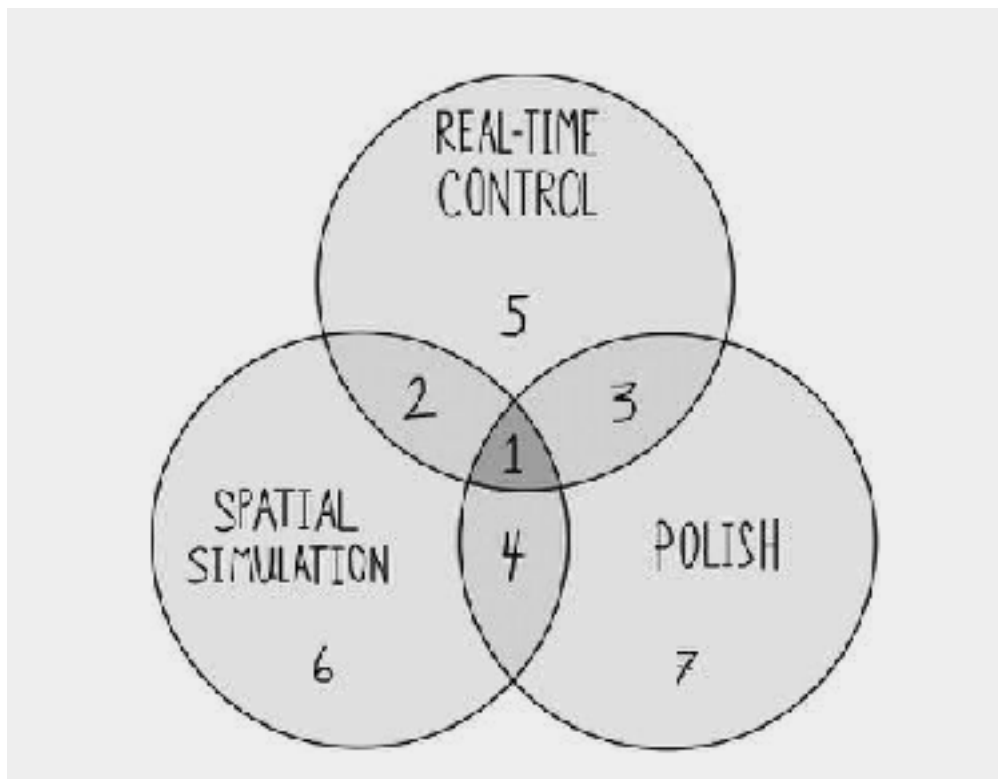
04.07.2017



Swink: Game Feel (2009)



1. **Input** -- How the player can express their intent to the system.
2. **Response** -- How the system processes, modifies, and responds to player input in real time.
3. **Context** -- How constraints give spatial meaning to motion.
4. **Polish** -- The interactive impression of physicality created by the harmony of animation, sounds, and effects with input-driven motion.
5. **Metaphor** -- The ingredient that lends emotional meaning to motion and provides familiarity to mitigate learning frustration.
6. **Rules** -- Application and tweaking of arbitrary variables that give additional challenge and higher-level meaning to motion and control.



Game Feel

aesthetic sensation of control

Polish refers to any effect that artificially enhances interaction without changing the underlying simulation. This could mean dust particles at a character's feet as it slides, a crashing sound when two cars collide, a “camera shake” to emphasize a weighty impact, or a keyframed animation that makes a character seem to squash and stretch as it moves. Polish effects add appeal and emphasize the physical nature of interactions, helping designers sell those objects to the player as real.

Swink: Game Feel, S. 5

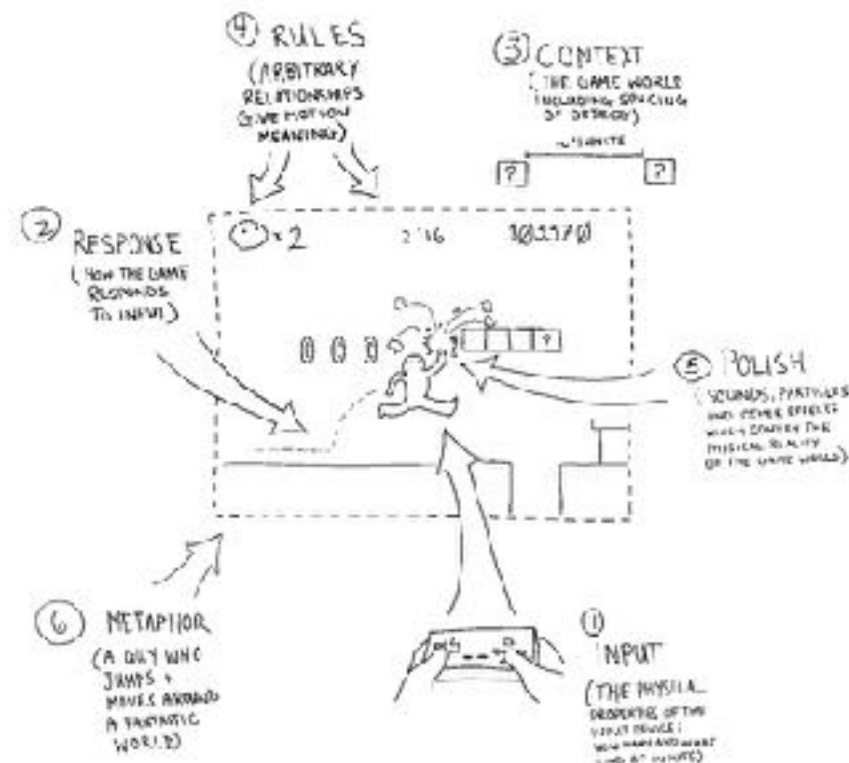
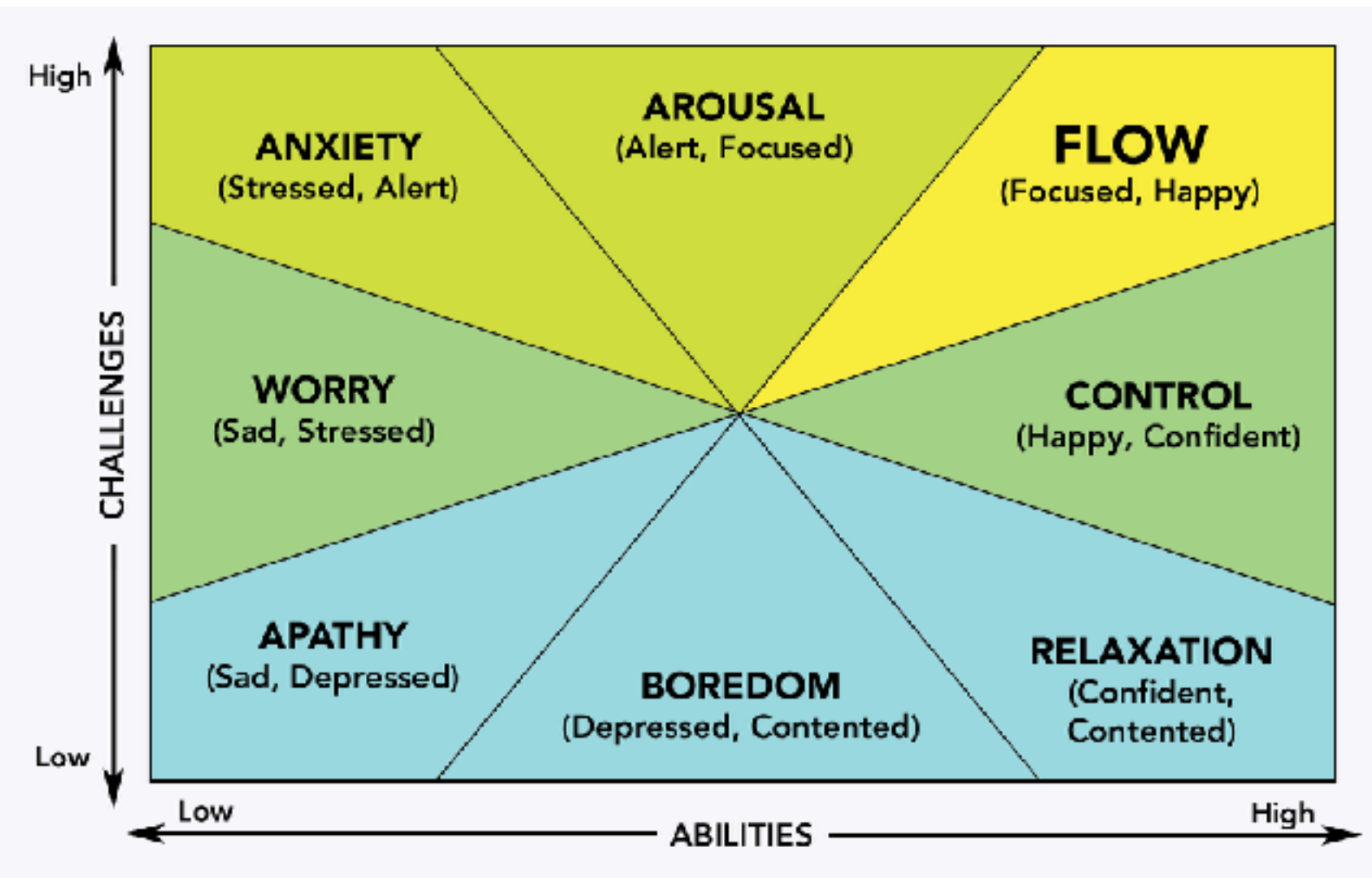


FIGURE 5.2 The six most important elements of game feel to measure are input, response, context, rules, polish and metaphor.

Game Flow

what people are talking about when they refer to the 'feel' of a game is its capacity to get us into a flow state.



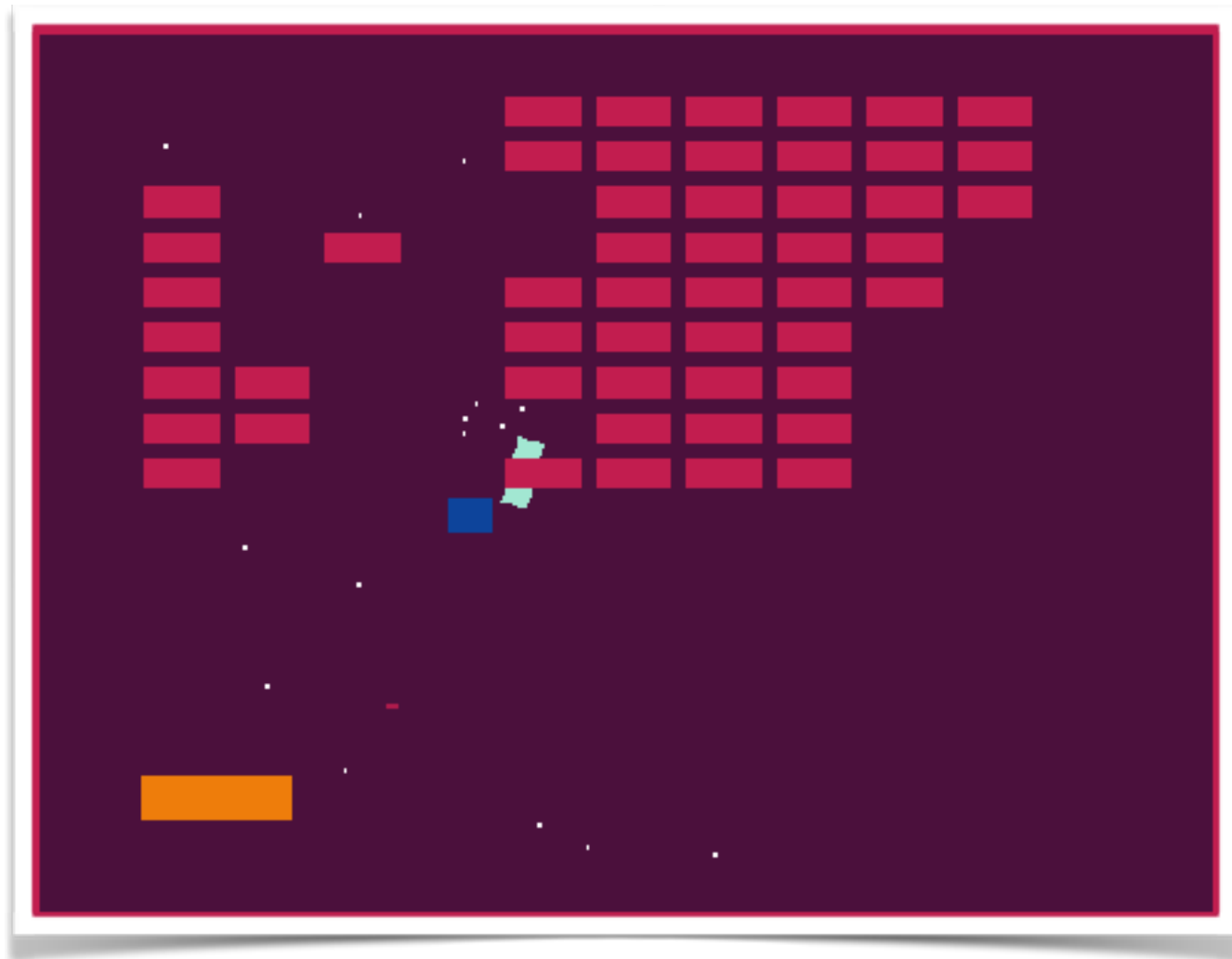
<https://www.rockpapershotgun.com/2007/11/25/touchy-feely>

Stand der Praxis

Juice it or Loose it

<https://www.youtube.com/watch?v=Fy0aCDmgnxg>

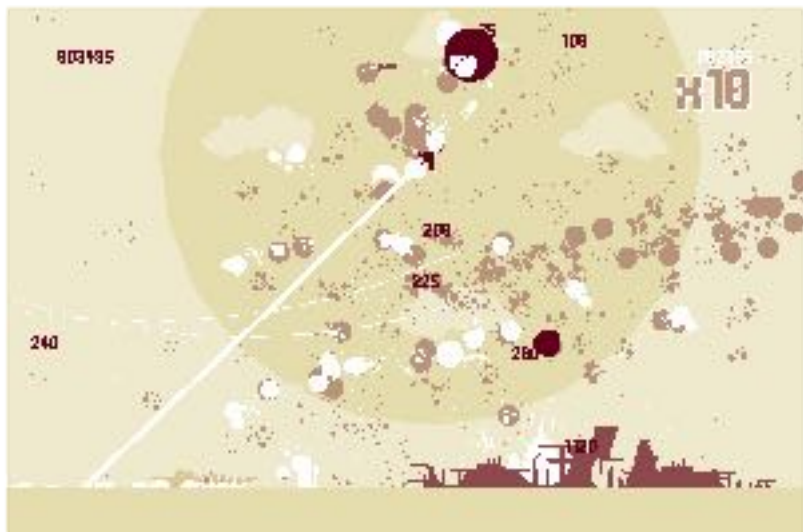
Martin Jonasson & Petri Purho (2012)



- Color
- Tweening / Easing (Position, Rotation, Scaling)
- Random Delay
- Paddle Stretch (Velocity)
- Ball Rotation, Stretch
- Animate Stretchin (Wobble)
- Color Change on Impact
- Scale Blocks on Impact
- Sound (Walls, Block, Paddle)
- Music
- Particles (Impact)
- Animation (Block Hit)
- Shatter
- Ball Trail
- Screen Shake
- Anthropomorphisation (Big Eyes, Blink, Smile)
- Screen Color
- Multiple Balls

<https://love2d.org/forums/viewtopic.php?t=77734>

<https://github.com/grapefrukt/juicy-breakout>



The Art of Screenshake

Jan Willem Nijman (2013): „Just fill your game with love and tiny details.“

<https://www.youtube.com/watch?v=AJdEqssNZ-U>

- Basic Animation and Sound
- Lower Enemy HP
- More Shooting / Enemies
- Bigger Bullets
- Muzzle Flash (First Frame is a Circle)
- Faster Bullets
- Less Accuracy
- Impact Effect
- Hit Animation
- Enemy Knockback
- Permanence (Corpses)
- Camera Lerp
- Camera Position
- Screen Shake
- Player Knockback
- Sleep (Pause Game on Hit)
- Gun Delay
- Strafe
- More Permanence (Bullet Shells)
- More Bass to SFX
- Super Machinegun
- Random Explosions
- Faster Enemies
- More Enemies
- Even higher rate of fire
- Camera Kick (Opposite Direction of Shoot)
- Bigger Explosions
- Even More Permanence (Smoke)
- Meaning (Game Over Screen)
- Slow Motion

Secrets of Game Feel and Juice Game Maker's Toolkit

Mark Brown (2015)

https://www.youtube.com/watch?v=216_5nu4aVQ



Super Time Force

- Hotline Miami (Dennaton Games, 2012)
- Random Heroes (Ravenous Games, 2012)
- Super Time Force (Capybara Games, 2014)
- Rogue Legacy (Cellar Door Games, 2013)
- Super Meat Boy (Team Meat, 2010)
- Castle Crashers (The Behemoth, 2008)
- The Binding of Isaac: Rebirth (Nicalis / Edmund McMillen, 2014)
- Super Mario World (Nintendo, 1990)
- Super Mario 64 (Nintendo, 1996)
- Nuclear Throne (Vlambeer, 2015)
- rymdkapsel (Grapefrukt, 2013)
- Ridiculous Fishing (Vlambeer, 2013)
- Gunbrick (Nitrome, 2015)
- Street Fighter II: The World Warrior (Capcom, 1992)
- God of War (Santa Monica Studio, 2005)
- The Legend of Zelda: The Wind Waker (Nintendo, 2002)
- Shantae and the Pirate's Curse (WayForward, 2014)
- Guacamelee (Drinkbox Studios, 2013)
- Shank (Klei Entertainment, 2010)
- Gun Godz (Vlambeer, 2013)
- Super Crate Box (Vlambeer, 2010)
- Peggle (PopCap Games, 2007)
- Alien Hominid HD (The Behemoth, 2007)

Bringing Your Game to Life in 10 Simple Steps

Sonny Bone (2015)

<https://gamedevelopment.tutsplus.com/tutorials/bringing-your-game-to-life-in-10-simple-steps--cms-23447>



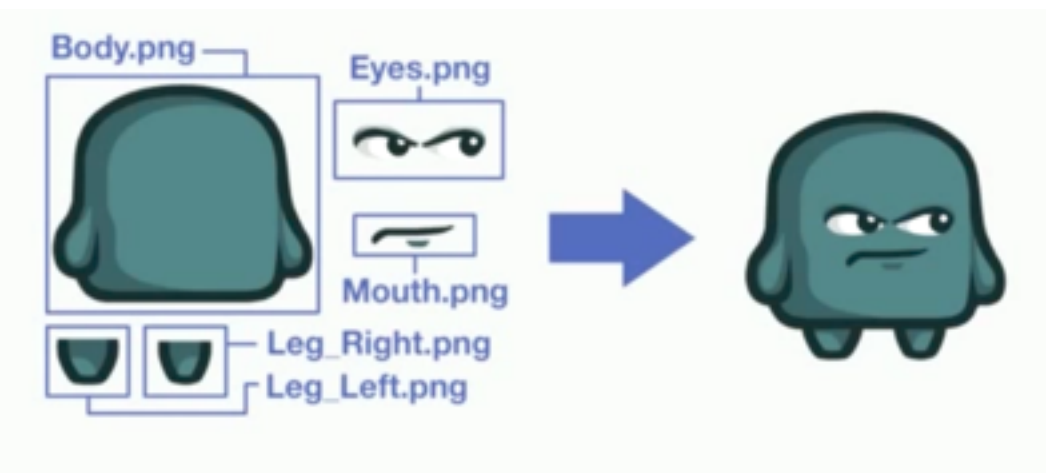
<https://www.youtube.com/watch?v=cpXKuF1vix8>

- 1: Basic Core Mechanics
- 2: Adding Animation
- 3: Adding Basic Sound Effects
- 4: Adding Stage Tiles
- 5: Verticality and the Introduction of Variation
- 6: Improving the Feel of Gunplay Through Feedback and Randomization
- 7: Enemy Knockback
- 8: Particles!
- 9: Enemy Death and the Value of Permanence
- 10: Final Touches

Why Your Death Animation Sucks



Move or Die



Nicolae Berbece (2015)

<https://www.youtube.com/watch?v=pmSAG51BybY>

- Animations (Sprite Sheets | Asset Sheets)
- Sound Effects (Movement | Death | Environment)
- Death Paint (Depth | Foreground Mask)
- Particles
- Screen Shake
- Chromatic Aberration
- Shockwave
- Haptic Feedback
- Props
- Landing Dust
- Eyes
- Rhythm (Background Animation)

Game Juice

adding layers of satisfying bits of animation and audio to improve its feel.

<http://www.gameanalytics.com/blog/squeezing-more-juice-out-of-your-game-design.html>



Jelly Jump

Animation

Tweening / Easing

Landing Dust

Pausing

Trails

Procedural Movements

Shader and Vertex Colors

Sparkles

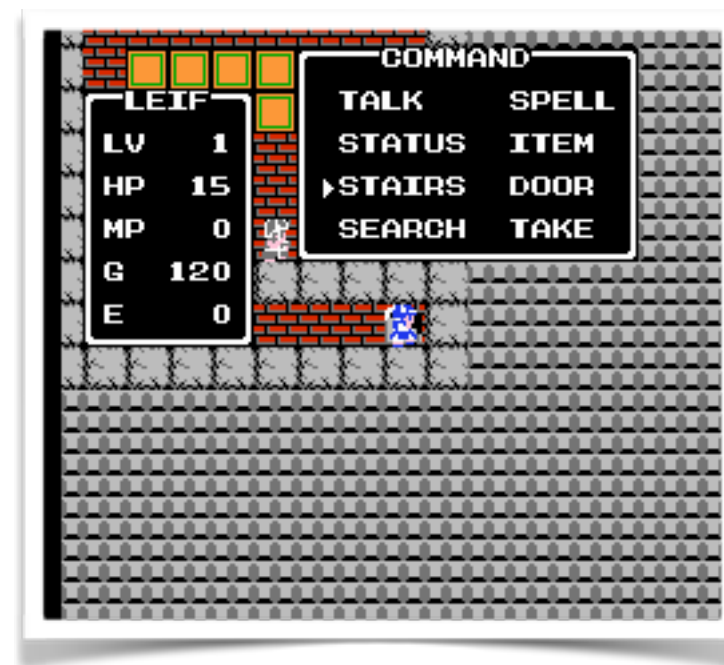
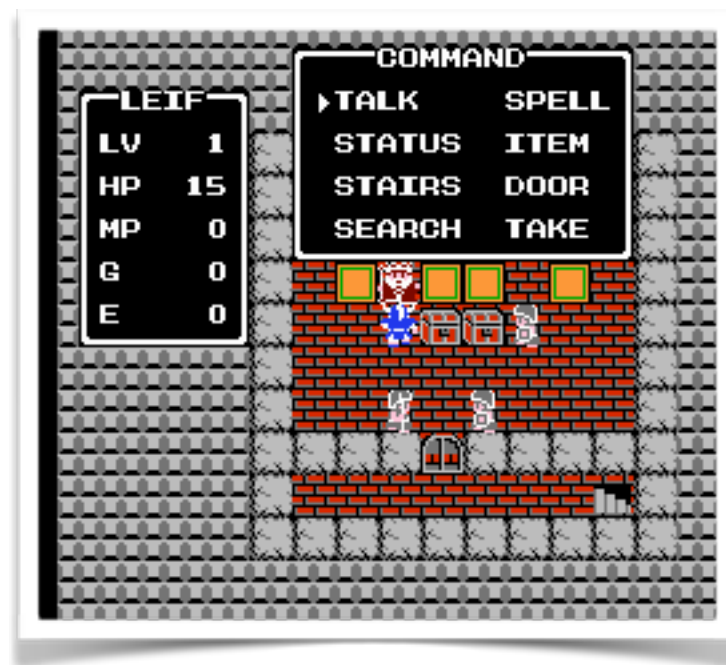
Audio

Music

Game Oil

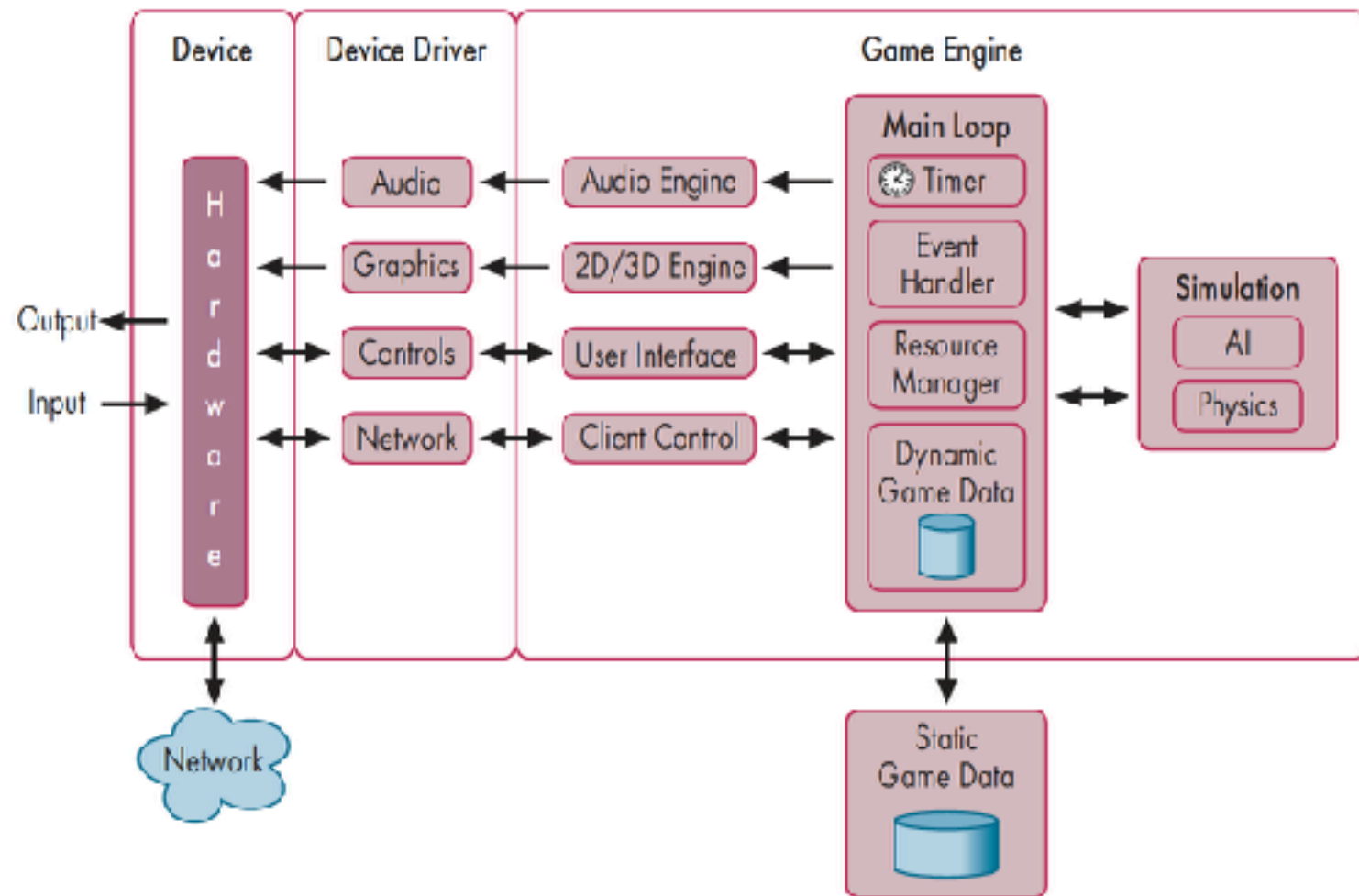
http://www.gamasutra.com/blogs/LarsDoucet/20160810/279009/Oil_it_or_Spoil_it.php

Juice adds pleasure, Oil removes pain.



If there is only one possible, unambiguous, non-regrettable action, the action button should just do that one thing.

Taxonomie



Game Juice kann in jedem Modul einer Game Engine hinzugefügt werden

Graphics

- Animation: Tweening, Blend
- Deformation: Stretch, Rotation
- Shader: Color, Chromatic Aberration
- Model: Anthropomorph, Following Eyes, Bigger Bullets
- VFX: Trail, Lens Flare, Halo
- Particles: Explosions, Smoke, Landing Dust
- Permanence: Death Paint, Corpses, Shells, Particles

Camera

- Follow
- Shake

Audio

- SFX
- Musik
- Bass

Time

- Pause
- SloMo

Physics

- Kollision: Tolerance
- Knockback
- Jump

AI

- Irrational Behaviour

Ludition

- Balancing
- No. Enemies
- Variation
- Randomization
- Triple Shoot
- Mechanics

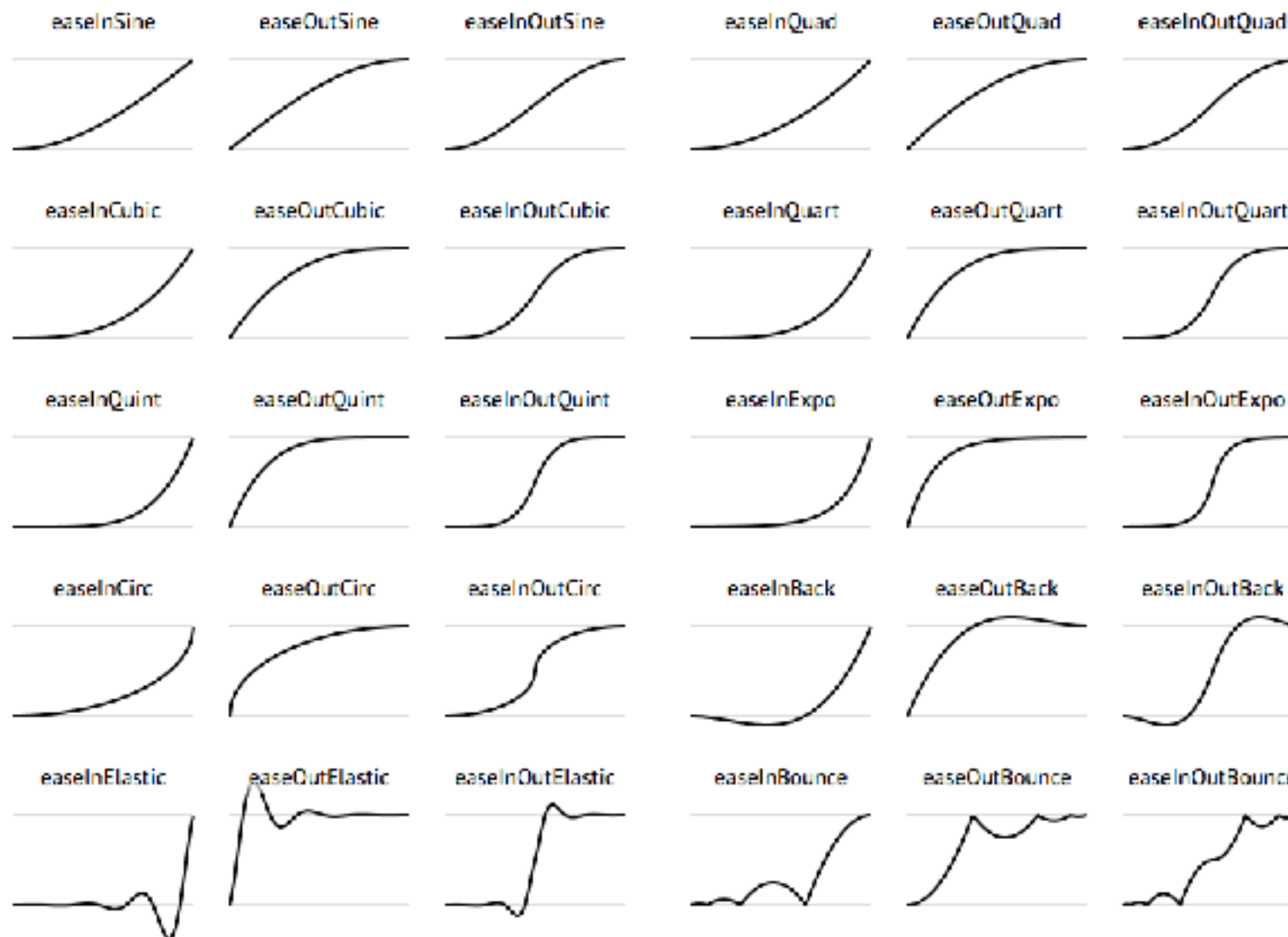
Controller

- ADSR
- Haptic Feedback

Animation

Tweening / Easing

<http://robertpenner.com/easing/>

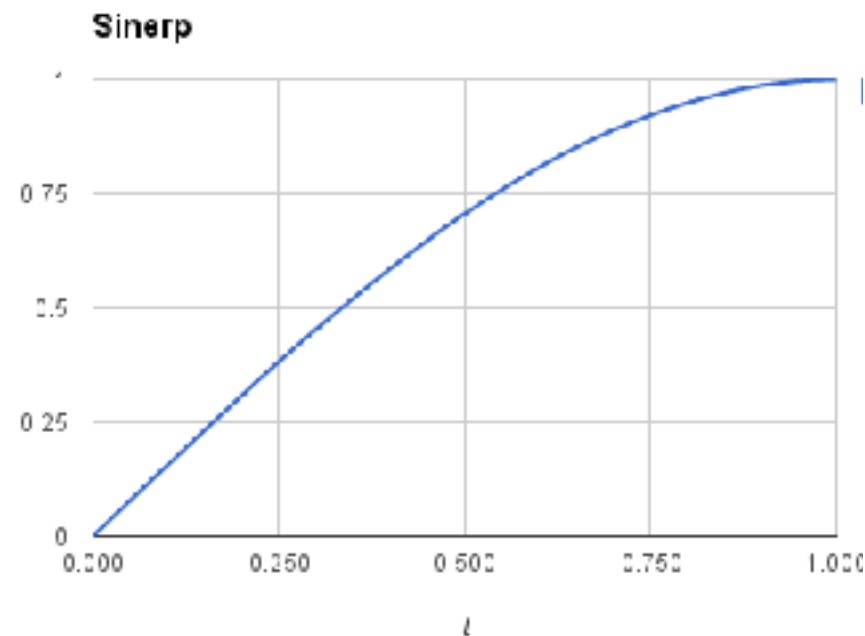


<http://easings.net/de>

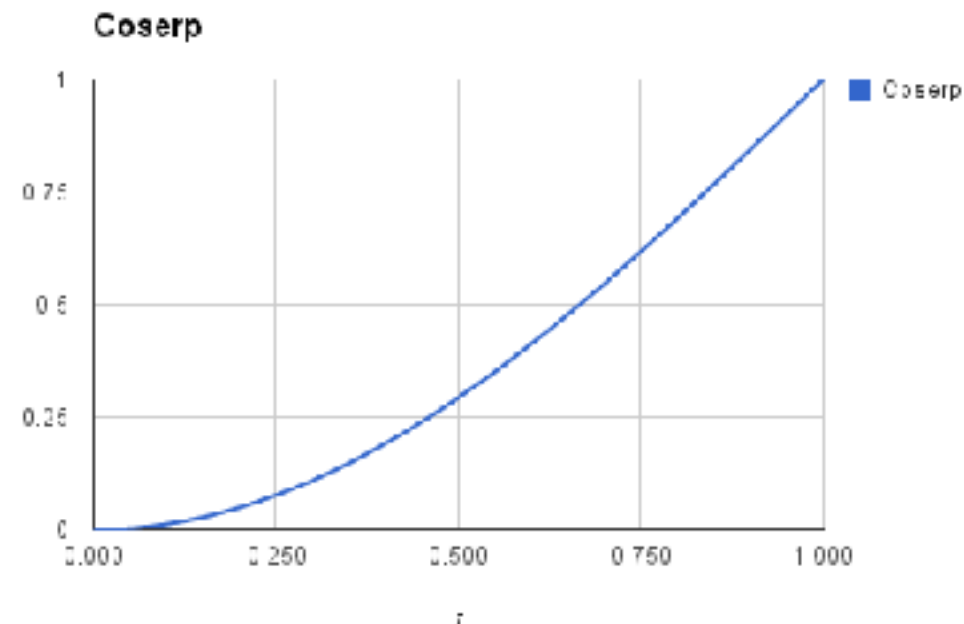
Tweening / Easing



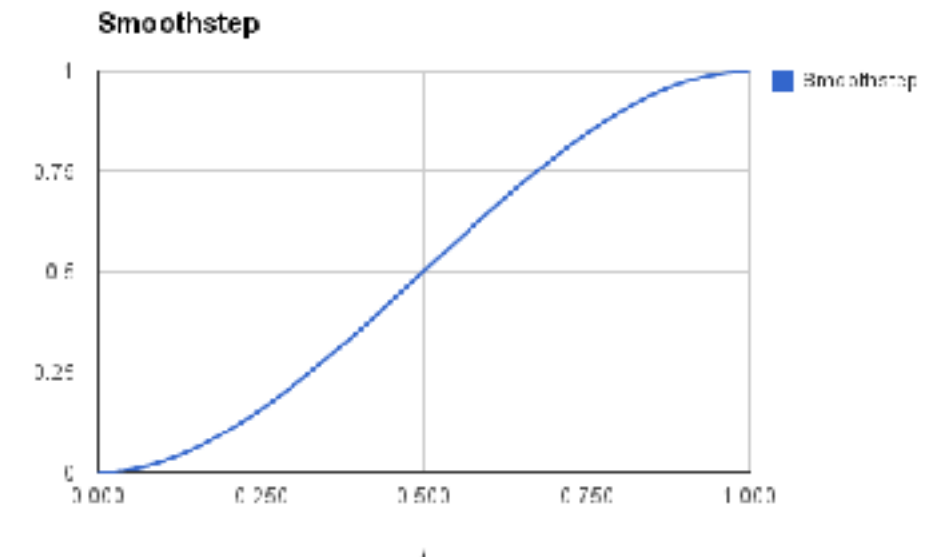
<https://chicounity3d.wordpress.com/2014/05/23/how-to-lerp-like-a-pro/>



```
float t = currentLerpTime / lerpTime;  
t = Mathf.Sin(t * Mathf.PI * 0.5f);
```



```
t = 1f - Mathf.Cos(t * Mathf.PI * 0.5f)
```



```
t = t*t * (3f - 2f*t)
```

<https://gist.github.com/Fonserbc/3d31a25e87fdaa541ddf>

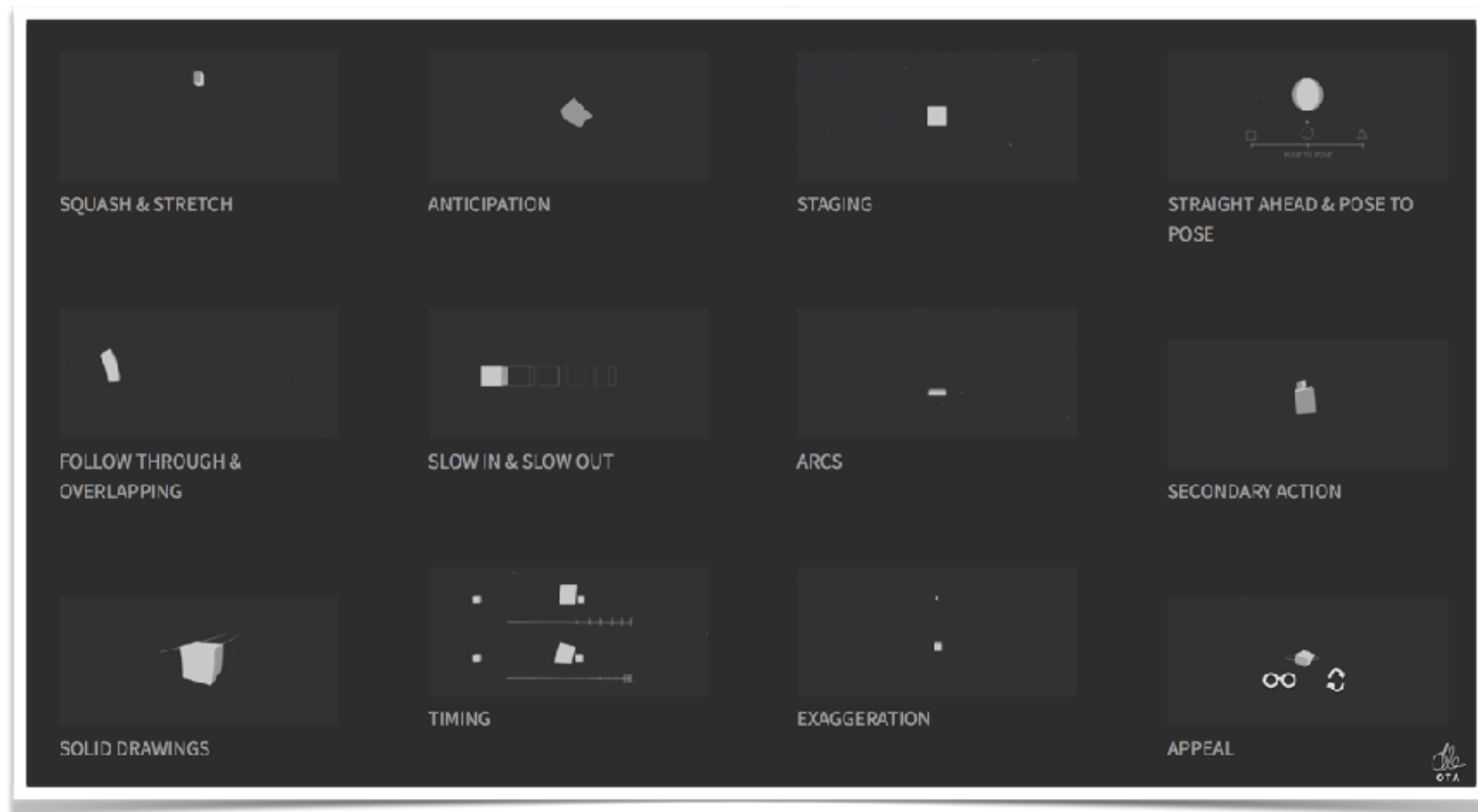
<http://www.pixelplacement.com/itween/index.php>

iTween <http://answers.unity3d.com/questions/976723/easing-curves.html>

<https://gist.github.com/cjddmut/d789b9eb78216998e95c>

<http://hotween.demigiant.com/getStarted.html>

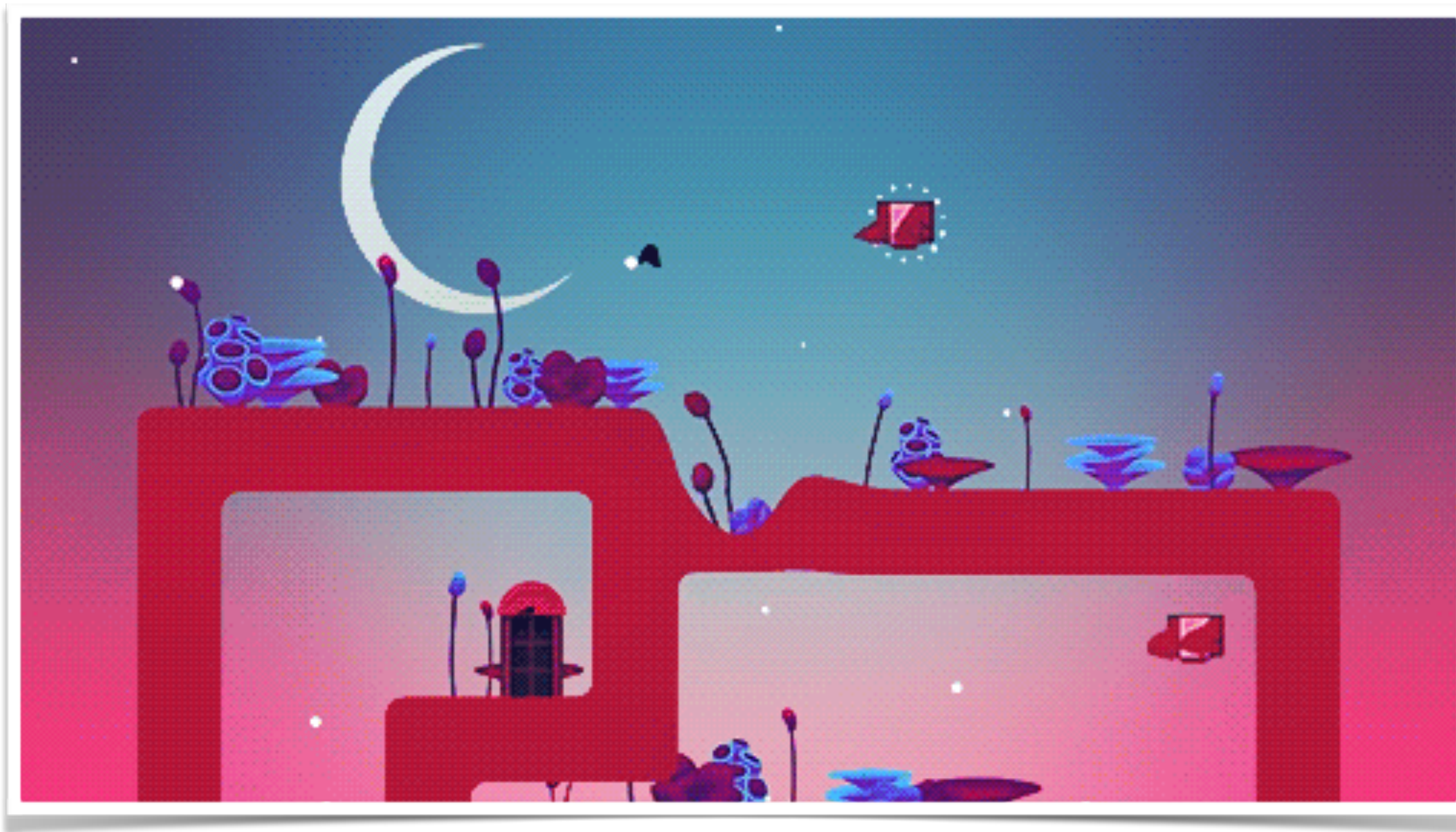
12 Principles of Animation



<http://the12principles.tumblr.com/>

<http://animationsmears.tumblr.com/post/113185794269/finally-writing-about-something-thats-been-on-my>

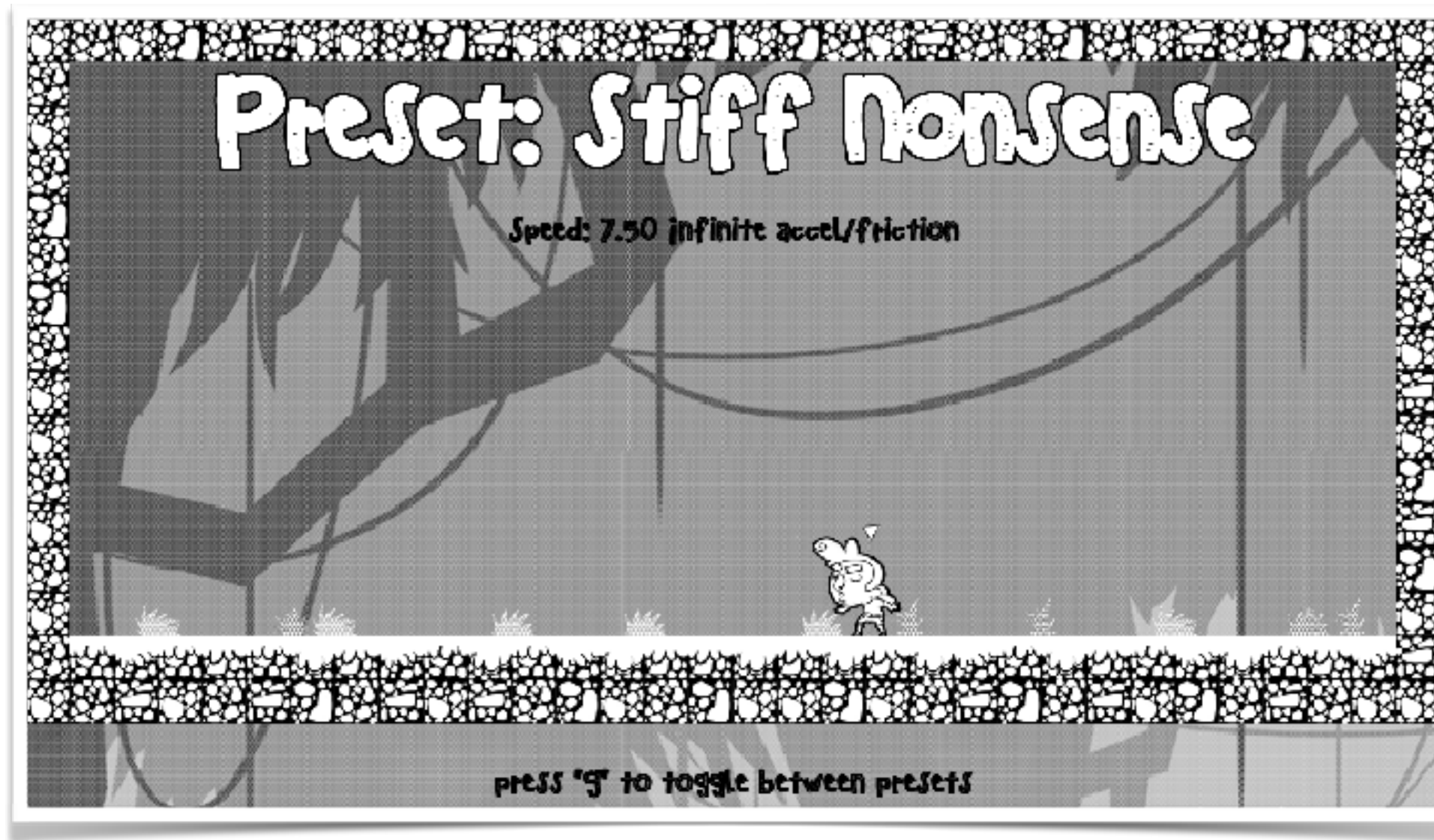
Responsiveness



The Floor is Jelly

<https://vimeo.com/83281882>

Jump



- “Stiff Nonsense”
- “Responsiveness”
- “Precision”
- “Floaty Nonsense”

Camera

Camera in Side-Scrollers

Itay Keren: Scroll Back: The Theory and Practice of Cameras in Side-Scrollers

Vortrag: <https://www.youtube.com/watch?v=pdvCO97jOQk>

Folien: http://www.gamasutra.com/blogs/ItayKeren/20150511/243083/Scroll_Back_The_Theory_and_Practice_of_Cameras_in_SideScrollers.php



Super Mario World Camera Logic Review

<https://www.youtube.com/watch?v=TCIMPYM0AQg>

Camera Mistakes

Vortrag: <https://www.youtube.com/watch?v=C7307qRmIMI>


Folien: <https://matellion.blogspot.de/2016/08/50-common-game-camera-mistakes-and-how.htm>

GAME DEVELOPERS CONFERENCE 2014

MARCH 17-21, 2014 GDCONF.COM

#4: Using a default camera distance that's likely to break line-of-sight.

- The smaller the danger zone, the easier it is to avoid.
- Line-of-sight must be small enough to fit through levels.
- Looming obstacles can intersect from the side, from behind, from ahead, or from below...



Screen Shake



```
void OnCollisionEnter2D(Collision2D coll)
{
    shakeAmt = coll.relativeVelocity.magnitude * .0025f;
    InvokeRepeating("CameraShake", 0, .01f);
    Invoke("StopShaking", 0.3f);
}

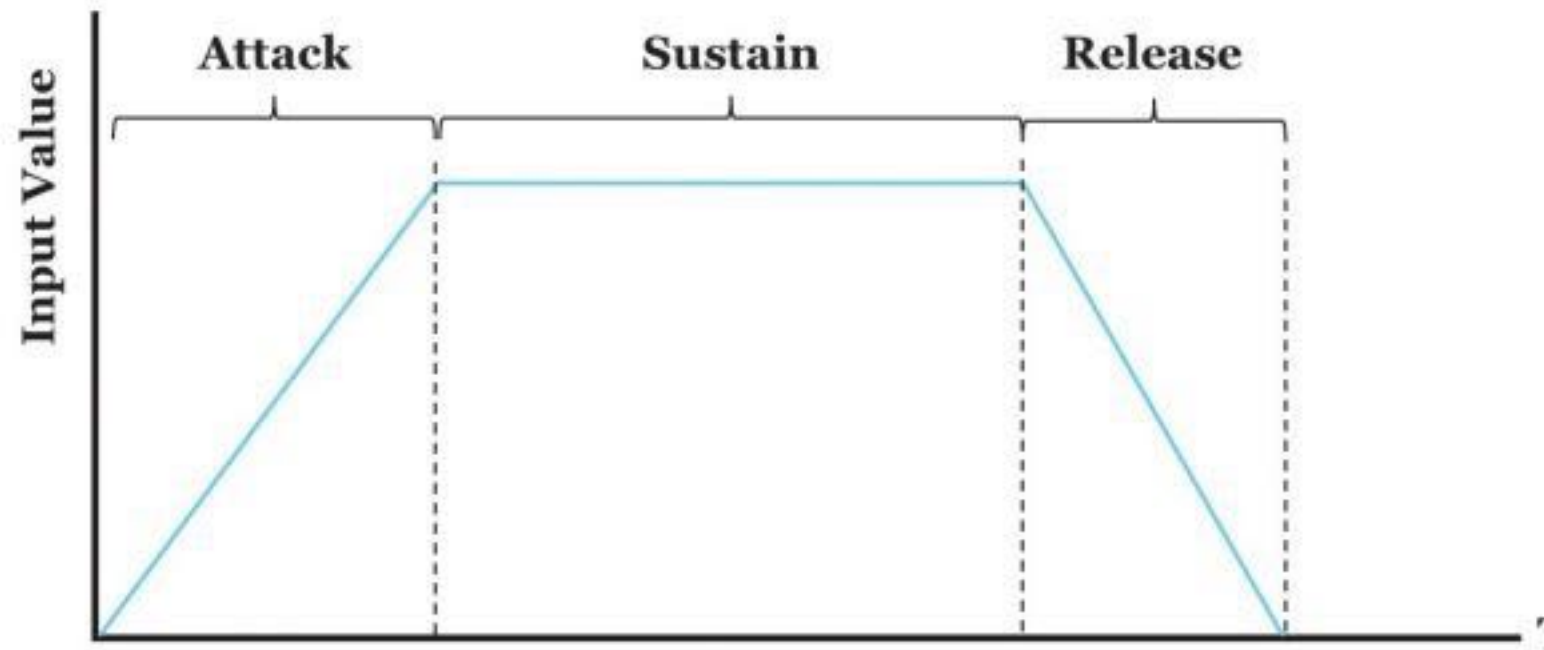
void CameraShake()
{
    if(shakeAmt>0)
    {
        float quakeAmt = Random.value*shakeAmt*2 - shakeAmt;
        Vector3 pp = mainCamera.transform.position;
        pp.y+= quakeAmt; // can also add to x and/or z
        mainCamera.transform.position = pp;
    }
}
```

<http://newbquest.com/2014/06/the-art-of-screenshake-with-unity-2d-script/>

Controller

ADSR

Curves – filter signal strength depends on time/speed.



On the left, you can see the gamepad input curve. To mimic this signal for the keyboard, we used an input curve that takes into account how long the steering button was pressed, which is allowed to turn the digital input from the keyboard into an analog signal. The curve is quite fast, with a very short delay time (the game has pretty arcade-style driving), but even such subtle mechanism makes control of the car much more natural and smoother.

http://www.gamasutra.com/blogs/AndrewDotsenko/20170329/294676/Designing_Game_Controls.php

Control Assists

predict the player intention and
help to perform desired actions



Designing Game Controls

http://www.gamasutra.com/blogs/AndrewDotsenko/20170329/294676/Designing_Game_Controls.php

Magnetism



Magnetism is a very powerful paradigm to use when we need the user to be precise. For example, consider a first-person shooter where the player needs to aim at enemies from far away in combat. At least in "easy" mode, it makes sense to code an auto-aim, so that if the enemy is located more or less in front of the player, it takes care of the precise aiming. This is what magnetism is about: making some positions dominate, so the user can have more fun with less effort. But sometimes magnetism can get in the way of realism. An advanced player will want to aim manually, so magnetism should always be an option, not a must.

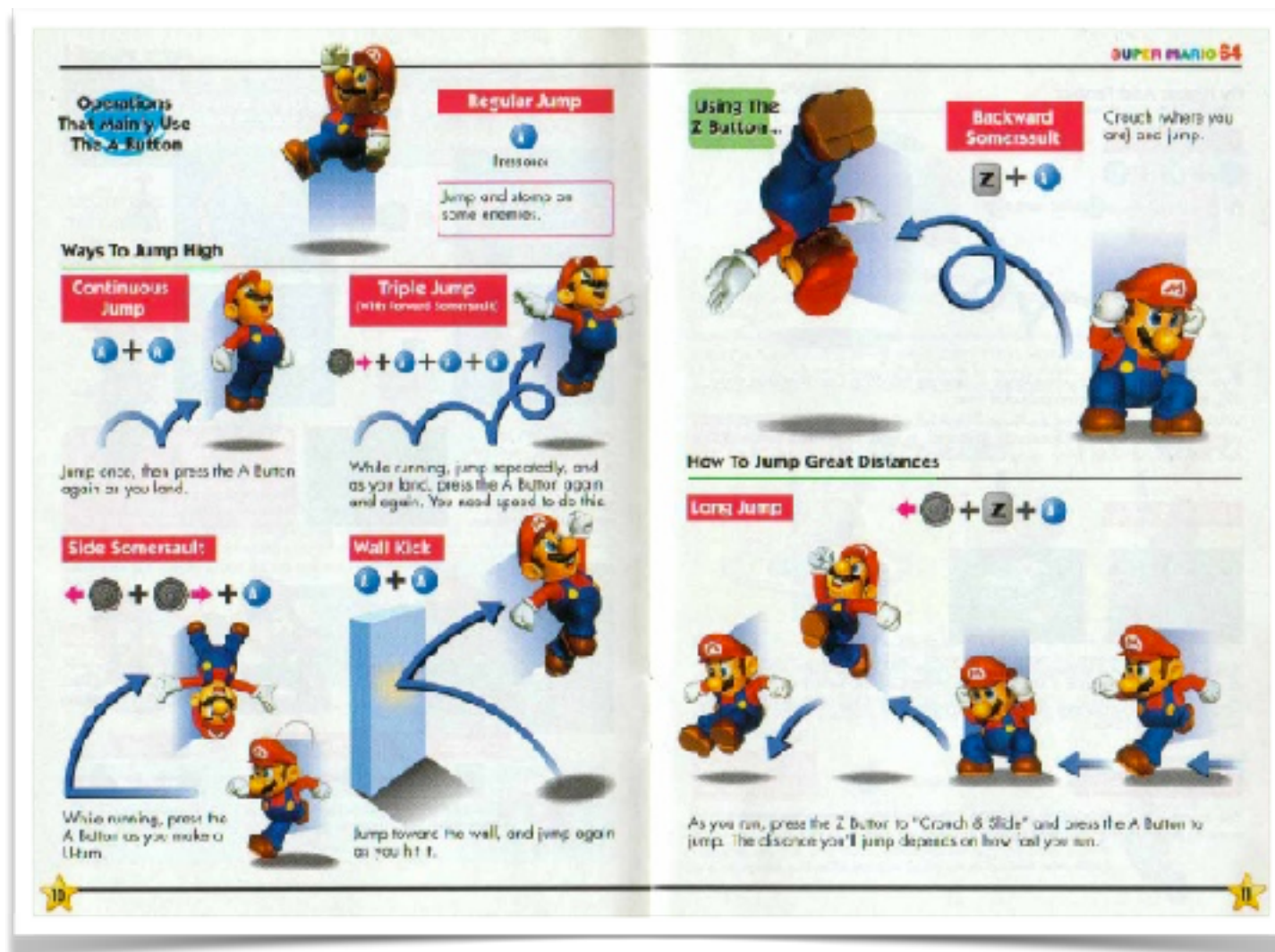
Another interesting use of magnetism is found in strategy games or, generally speaking, games where we must be able to pick units or positions on a map. In a strategy game, we need our mouse cursor to gravitate around units so it is easier to select them. Imagine that you have one unit selected and are in "combat" mode, so you must click on the position of the map where the unit should move in order to engage in a fight. Because it doesn't make sense to fight against a rock or a piece of land, you can code the GUI so when the user clicks, it automatically detects the closest unit to the cursor and sends the player unit there. This way we can prevent the inexactitudes in user control that make the game less fun.

Dual Stick Control



Mario's Jump

- The Secret of Mario's Jump (and other Versatile Verbs)
Game Maker's Toolkit <https://www.youtube.com/watch?v=7daTGyVZ60I>



- Button: Press, Hold, Release,
- Cooldown Time
- Multipress (Double Jump)
- Combo (Multiple Buttons) (State Machine)
- Modify another Verb (Aim)
- Modify with Analog Input (Velocity)
- Combining Verbs

Expression Space

Designing Game Controls

http://www.gamasutra.com/blogs/AndrewDotsenko/20170329/294676/Designing_Game_Controls.php

Expression instead of learning

Hard to master

Different skill level

Control states

Skill Combinations

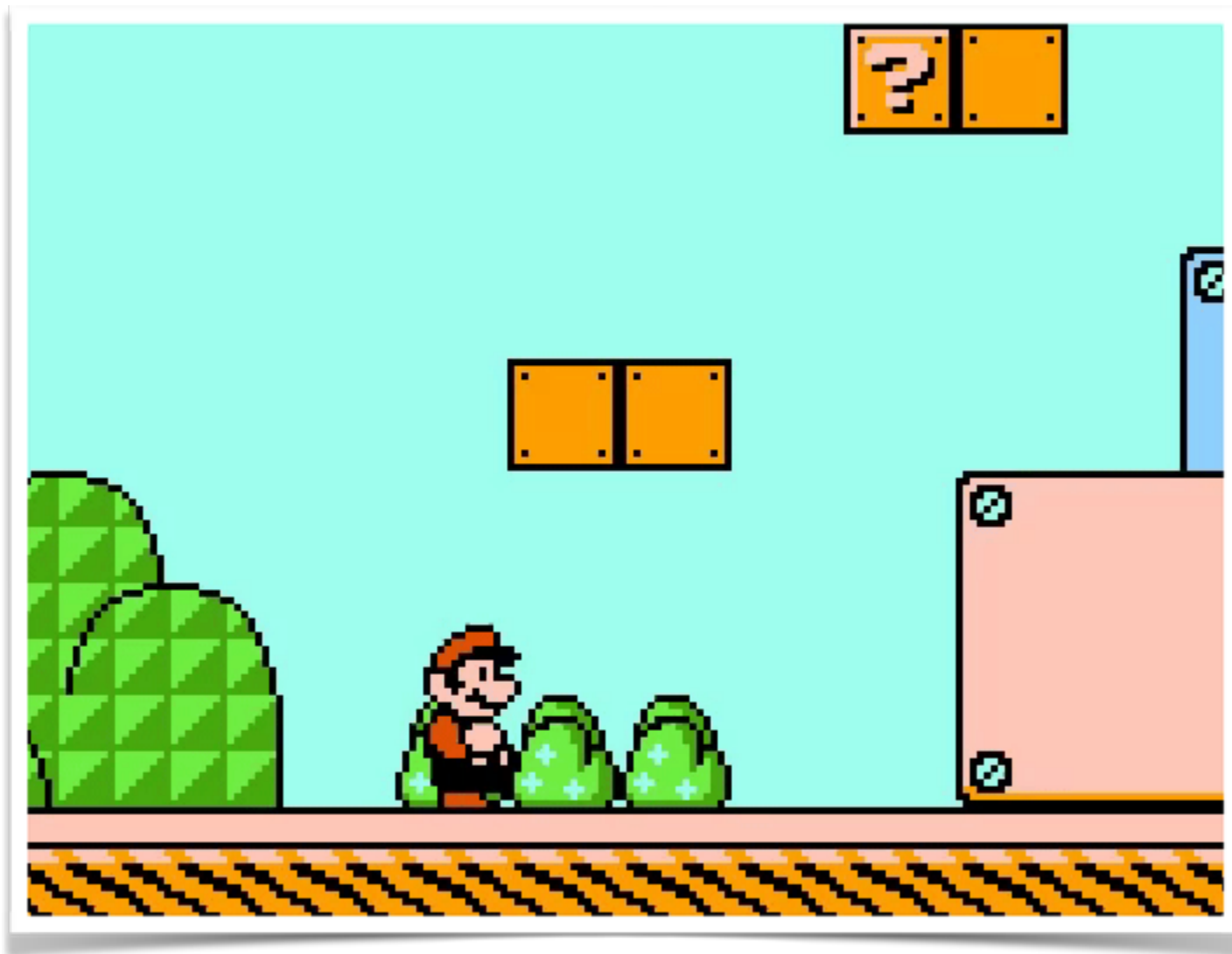
Different goals and context



+ high level of execution

Physics

Collision Tolerance



- Graceful platform edges
- smoothing off the sharp corners!
- Insgesamt: Toleranzen des Systems erhöhen

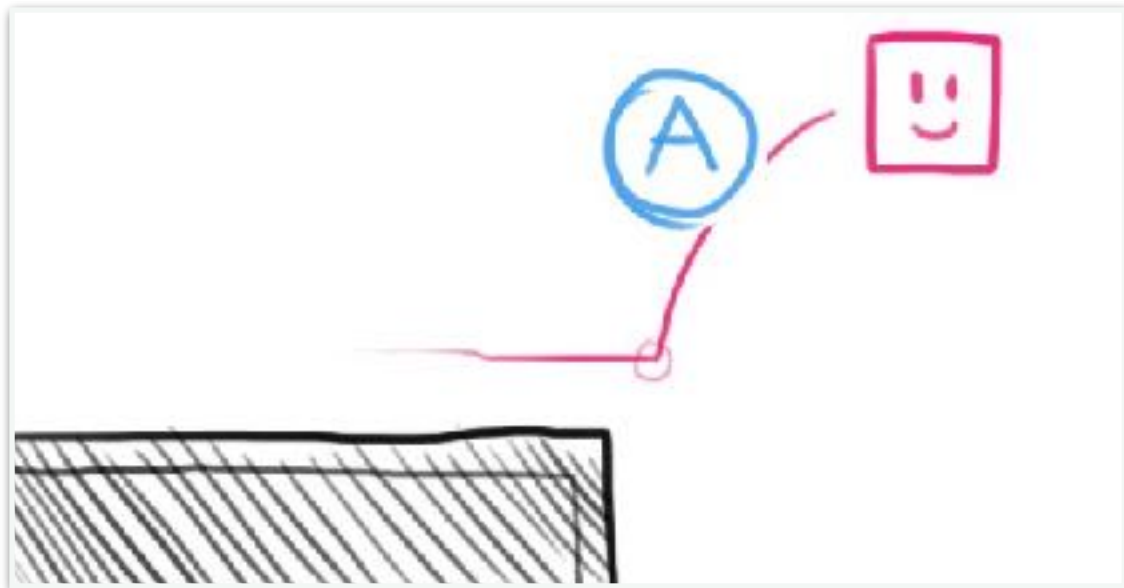
Platform Ledge Forgiveness

Kyle Pulver: http://kpulv.com/123/Platforming_Ledge_Forgiveness/

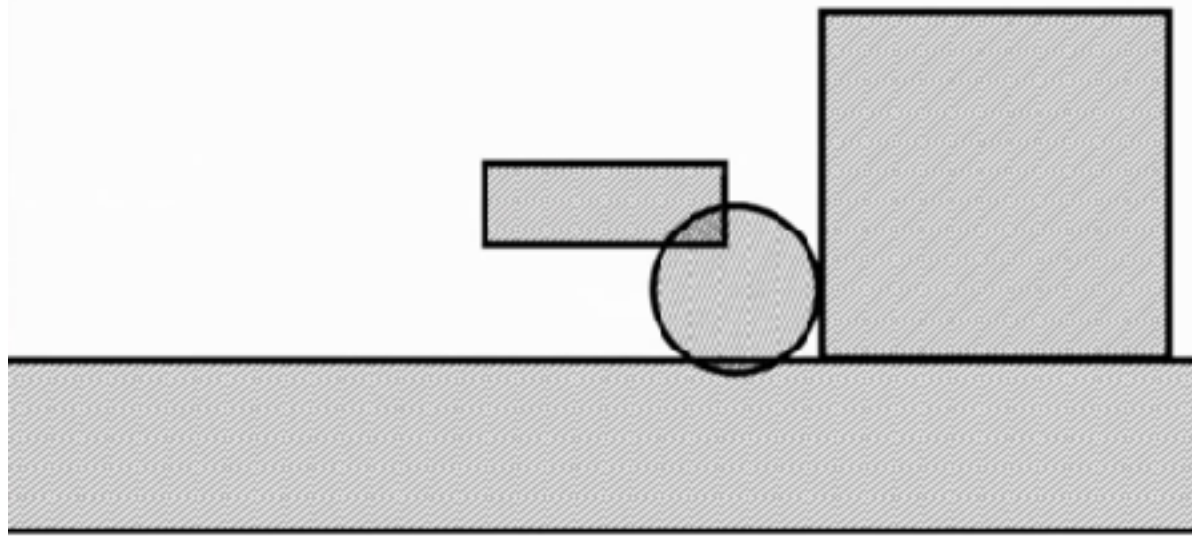


```
if (jumpInput) {  
    if (onGround || graceTimer > 0) {  
        jump();  
    }  
}
```

```
//somewhere else in the code:  
public function jump():void {  
    //pretend theres more jumping logic here  
    graceTimer = 0;  
}
```



Non-resolvable scenarios

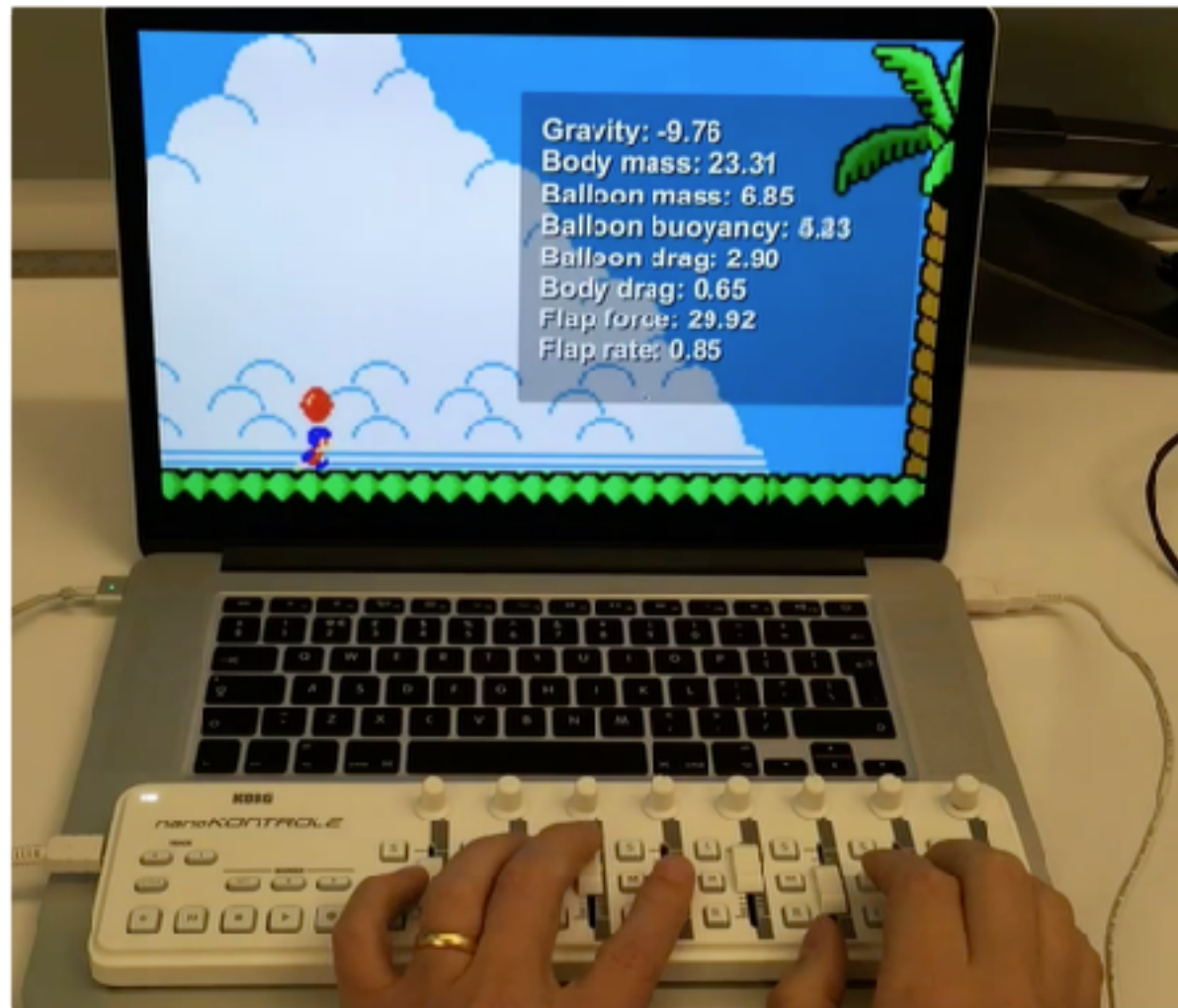


Physics

Bennett Foddy

Designing with Physics: Bend the Physics Engine to Your Will

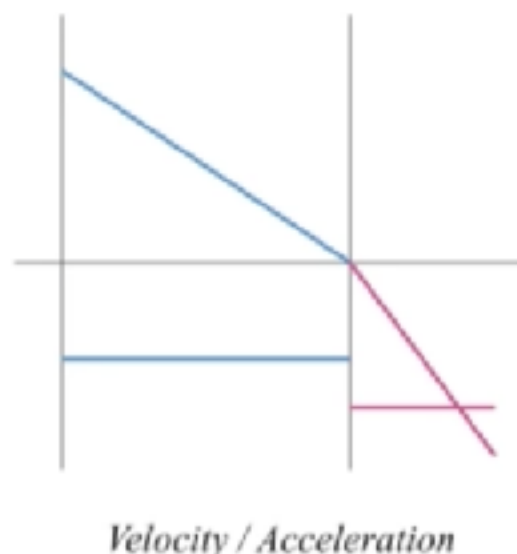
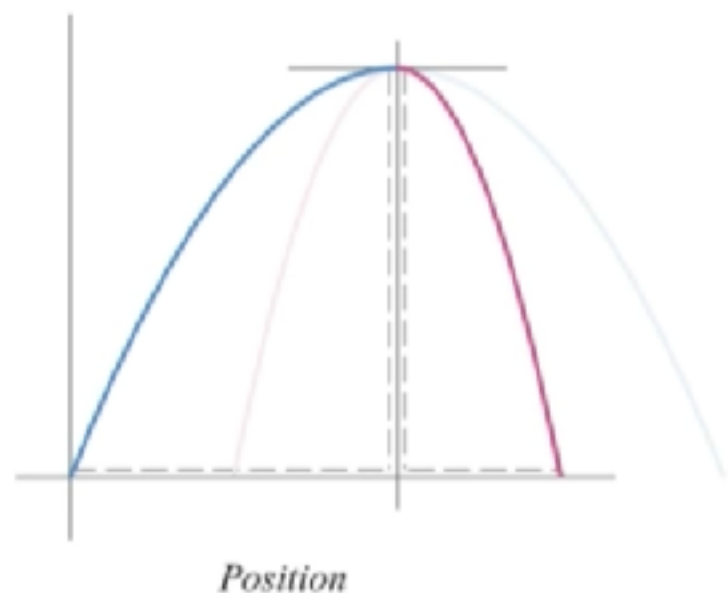
- Floaty vs Stable Physics
- Gravity, Units, Timesteps
- Mass values, Iterations, Non-resolvable scenarios, Drag Force



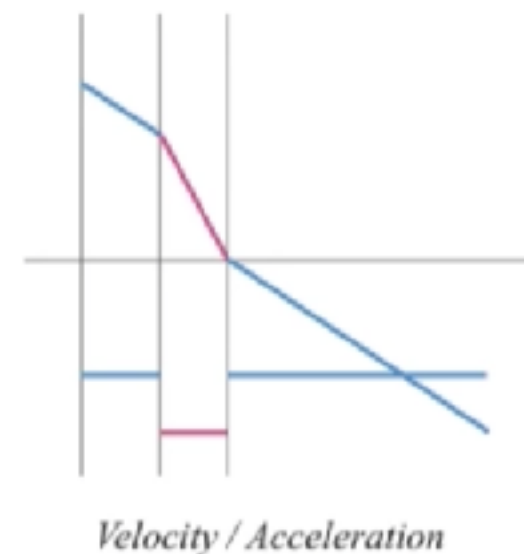
<http://www.gdcvault.com/play/1021921/Designing-with-Physics-Bend-the>

Building a Better Jump

Fast falling



Variable height jumping

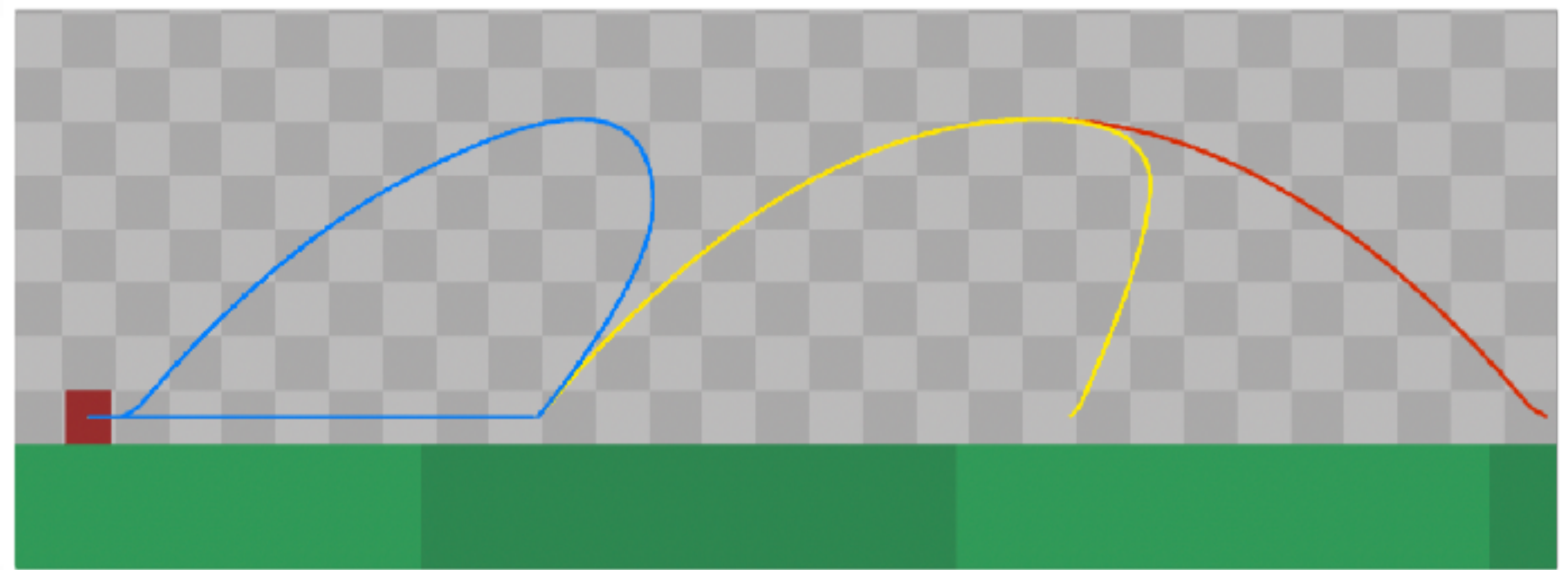
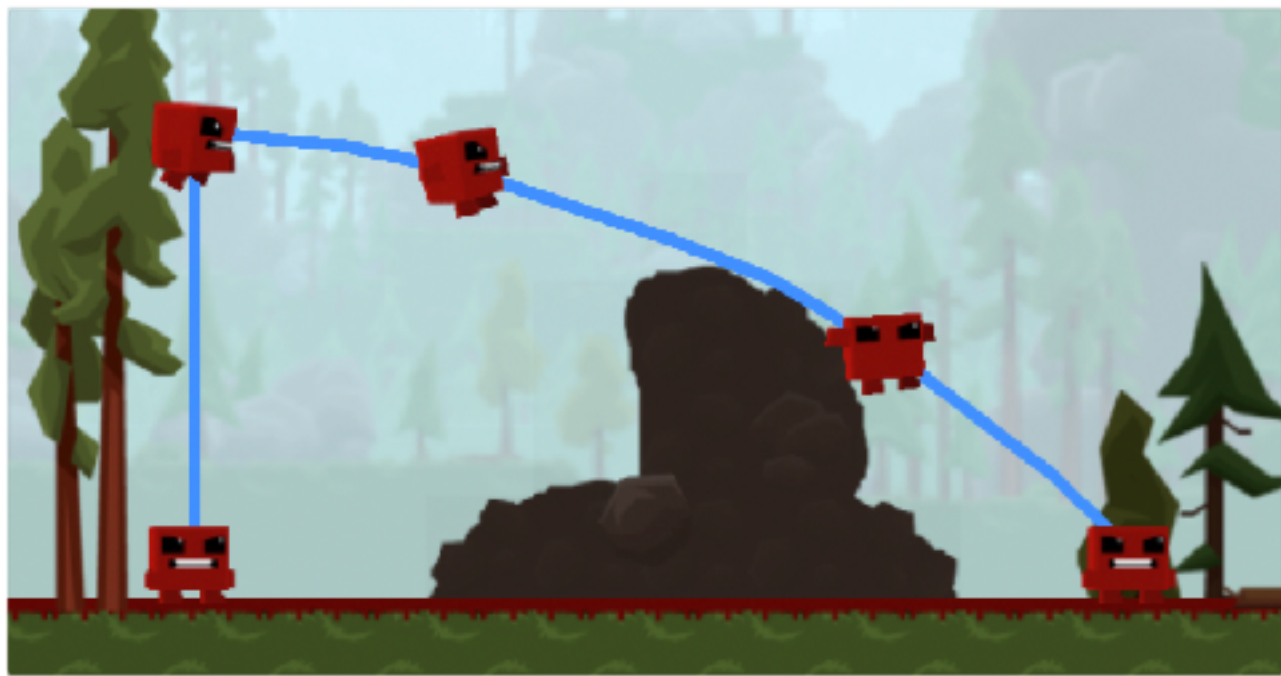


In this 2016 GDC talk, Minor Key Games' Kyle Pittman shows how to construct natural-feeling jump trajectories from designer-friendly input like desired height and distance, modeled programmatically using one of a few available integration methods.

<https://www.youtube.com/watch?v=hG9SzQxaCm8>

You Say Jump, I Say How High?

Martin Fasterholdt (2016)



<http://martinf.dk/jump/>

Mechanics: Combat Systems



God of War



Bayonetta

http://www.gamasutra.com/view/news/261698/7_combat_systems_that_every_game_designer_should_study.php

http://www.gamasutra.com/blogs/JamesMargaris/20150505/242687/Dark_Souls_Its_like_an_NES_Game.php

Kontraindikation

Juice ohne Kontext kann Immersion zerstören



Walking Dust in der Luft

Tweening hard elements (Rock felt like rubber)

Grading and dynamic lighting on pixel art

< Shadows in the Sky

<http://www.gdcvault.com/play/1020861/Don-t-Juice-It-or>

Make your game feel juicy with Unity

<https://www.youtube.com/watch?v=WfwRBwNz2bg>



Beim Versuch, die Vorträge von Nijman und Jonasson & Petri Purho zu imitieren, zeigt Arturo Núñez (unabsichtlich), wie diese Effekt *nicht* eingesetzt werden sollten.

- Spawn Tween (Easing)
- Spawn Tween Noise
- Cam Shake
- Fireburst (Particles)
- Movement Scale
- Destroy Particles
- SFX
- Randomize Shoot
- EnemySmoothMove
- TrailRenderer
- EnemyPersist (Fall)
- Bloom Intensity