CHAPTER 5

Mechanical Experience, Competency Profiles, and Jutsu

SASHA SORAINE AND JACQUES CARETTE

ABSTRACT

We look at players' mechanical experience of gameplay challenges. We consider mechanical experience as fundamental design knowledge: designers need to analyze the challenges they craft to understand the skills necessary from their players for success. One application of this study is to explain why some challenges may be inaccessible to certain players. There is currently no comprehensive framework for describing accessibility of challenge experiences. We also propose *challenge jutsu*, structured representations of challenge descriptions (via competency profiles) and player models, as a first step towards better understanding the mechanical profile of various game challenges and sources of difficulty.

INTRODUCTION

Different types of experience

Many have tried to understand the player's experience of a game. One method is to model the components of the game and relate them to "experiences." For example, the Mechanics-Dynamics-Aesthetics Framework and associated Eight Types of Fun explores this (Hunicke, LeBlanc, & Zubek, 2004). Schell (2014) uses an Elemental Tetrad to model and describe games as having four types of components: mechanics, aesthetics, story, and technology. These views segregate game components into the interactive parts created by designers and the subjective aspects open to interpretation by the players. This dichotomy seems artificial.

We view games as designed systems of experience; designing games is an exercise in crafting holistic experiences. Games can be experienced *mechanically* (through gameplay actions), *aesthetically* (through the visual and audio design), *emotionally* (through the narrative and characters), *socially* (through the communities of players), and *culturally* (through a combination of cultural interpretations and interactions). The experience of a game is a combination of these *aspects of experience*. Table 1 maps them to the Hunicke et al.'s (2004) Eight Types of Fun.

Table 1

Mapping Hunicke et. al Types of Fun to Experiences

Kind of fun	Definition of fun	Experience Type
Sensation	Game as sense-pleasure	Mechanical and aesthetic
Fantasy	Game as make-believe	Aesthetic, emotional, and socio-cultural
Fellowship	Game as social framework	Socio-cultural
Narrative	Game as drama	Emotional and socio-cultural
Challenge	Game as obstacle course	Mechanical
Discovery	Game as uncharted territory	Emotional and socio-cultural
Expression	Game as self-discovery	Emotional and socio-cultural
Submission	Game as pastime	Mechanical and aesthetic

We can visualize these experiential modes similarly to the Elemental Tetrad (Schell, 2014), with all aspects of experience being able to interact with one another. Whereas the Elemental Tetrad shows the parts of games, we propose the *Experiential Tetrad* to delineate the aspects of experience. These aspects reflect how the individual's abilities and knowledge relate with the game elements. *Mechanical experience* comes from interacting with the gameplay. *Aesthetic experience* comes from understanding and reacting to the aesthetic components. *Emotional experience* comes from relating to the story and characters. *Social and Cultural experience* comes from the player's personal interpretations of the game and their interactions with other people. The actual experience depends on the individual's point of view, context, interactions, and relation to the game.

We organize the Experiential Tetrad to show that different viewpoints change the experience (Figure 1). *Designers* have a comprehensive viewpoint, as they must craft and balance all aspects to create the experience of their game. *Observers* mainly witness the aesthetic, emotional, and socio-cultural aspects of games; while they may have an abstract understanding of the mechanics of the game, they do not experience the game mechanically unless they actually play the game. *Players* often first experience the mechanics, and then the other aspects after learning the mechanics. We focus on the player and designer viewpoints as they relate to experience design.

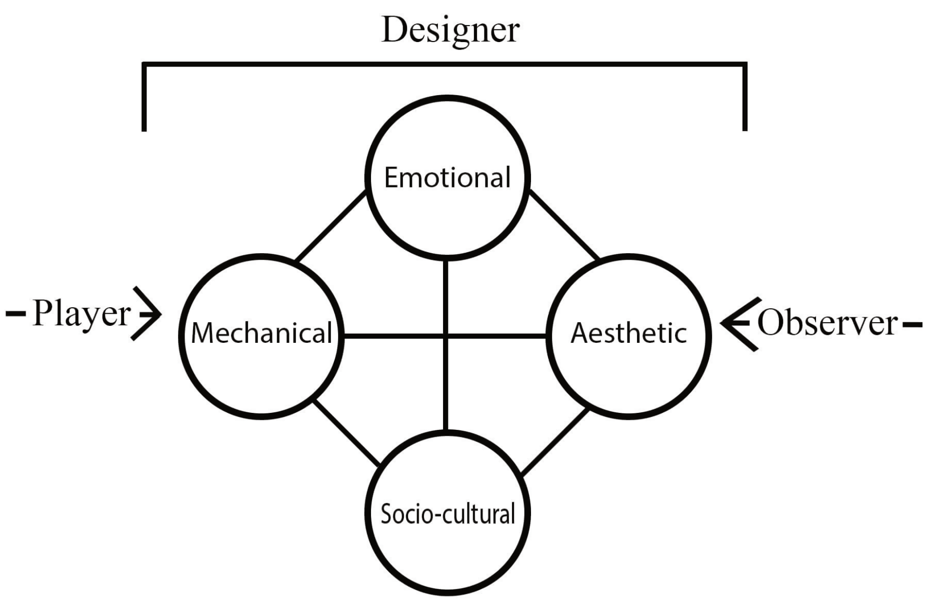


Figure 1: Experiential Tetrad. This figure illustrates the player experience as a tetrad with player, observer, and designer viewpoints.

For designers, crafting a game relies on understanding and managing how the aspects of the game relate to each other. Designers often have a primary aspect in mind when crafting an experience. The other aspects are tuned to emphasize or support that primary aspect. Consider horror games: the crafted experience is primarily emotional, supported by aesthetic and socio-cultural experiences. Mechanics must be simple enough, so they don't distract the player from the atmosphere the developer has crafted. Games like *Silent Hill: Shattered Memories* (Climax Studios, 2009), and *Amnesia: The Dark Descent* (Frictional Games, 2010) are good examples: the control mechanics are simple navigation and object interaction. This simplicity makes it easy to play while also making the player feel vulnerable and limited. Complex mechanics would distract from the other aspects, stealing too much of the player's attention. For expert players of games like *Five Nights at Freddy's* (Cawthon, 2014) or *Resident Evil 7: Biohazard* (Capcom, 2017), the emotional experience becomes secondary to just beating the game. Unlike other forms of entertainment, no matter which aspect of experience games emphasize, there is always an underlying mechanical experience that impacts the reception of the other aspects.

For players, experiencing what designers have crafted depends on successfully engaging with the mechanical experience of the game. Players use in-game actions to complete gameplay challenges. Executing these actions and interacting with the challenges requires various cognitive and motor abilities. Successful interactions require that the players' abilities match up with what the game is asking them to do. If players are unable to complete challenges due to a misalignment of abilities, they are often unable to progress in the game. Lack of progress will limit how they can experience the emotional, aesthetic, and socio-cultural aspects of the game.

This barrier to engagement should be seen as a fundamental problem in game design. Thus, we need to systematize our understanding of the mechanical experience of players to make *mechanically achievable gameplay*.

What can we do?

The *mechanical experience* is a relation between the player's physical and cognitive abilities and the challenges within gameplay. It is crafted by the designer through the selection of challenges. Challenges and player abilities combine to create the mechanical experience based on whether the challenge is achievable or not. The mechanical experience must be interesting (in and of itself) to the target audience or invisible when it is meant to be a supporting actor in the overall experience. Currently, there is no framework for designers, players, and critics to talk about this, which is where we begin our work. We need to understand the relationship between the mechanical experience *as designed* and as *experienced by* the players.

We view mechanical experience design as a task modeling problem. Games have various gameplay challenges that the player must successfully complete to win (Adams, 2010). Modeling gameplay challenges as tasks, we can then describe them in terms of the abilities that players need to successfully complete them. This fits with Fleishman, Quaintance, and Broedling's (1984) idea of a *competency profile*, the set of cognitive and motor abilities that characterize a task. Identifying competency profiles would let us to model the *expected mechanical experience* as intended by designers. From there we can model *actual mechanical experience* by how the player's abilities compare to the ones required to complete the tasks. This will let us pinpoint which abilities are the limiting factor in completion of a given challenge. This lets use be concrete about *mechanical difficulty source*. By understanding expected versus actual experience and mechanical difficulty sources, we can design more mechanically achievable gameplay.

To create the vocabulary and framework of mechanical experience, we need a list of human cognitive and motor abilities along which individuals can be evaluated, and a list of archetypal gameplay challenges which are mutually exclusive along their intrinsic competency profiles as defined by those abilities to discuss. To this end, we assembled the required information into a structure that we call a *challenge jutsu*[1], to organize

1. Jutsu meaning method, technique, art, or skill in Japanese; used in different martial arts

challenge competency profiles and player profiles, and to show the impact that small changes in game mechanics has on each. The challenge jutsu lets us to predict how the *mechanical experience* will change for different player profiles. The effectivity of challenge jutsu crucially relies on understanding the *mechanical difficulty source* of each challenge. Challenge jutsu double as a design and critiquing tool for comparing the experience of a challenge by different demographics; thereby allowing us to spot unintentional sources of difficulty due to a mismatch of abilities. We first delve more deeply into player abilities and competency profiles before exploring our *Jutsu Framework*.

THE PLAYER

Players are incredibly complex to model. Player typologies are common in game studies to understand motivations for play and player satisfaction. The main idea of player typologies is to identify player archetypes based on sets of psychometric (Bateman & Boon, 2005; Bateman, Lowenhaupt, & Nacke, 2011; Stewart, 2011; Tseng, 2010; Zackariasson, Wahlin, & Wilson, 2010) or in-game behavioural characteristics (Bartle, 1996; Drachen, Canossa, & Yannakakis, 2009; Yee, 2007). Analyses and criticisms on the usefulness and validity of typologies are many (e.g. Bateman et al., 2011; Hamari & Tuunanen, 2014). Despite their utility in other contexts, these play typologies do not suit our purposes as they focus on understanding why players make decisions in games, not how they may physically interact with a game. Our goal is to model the psycho-motor aspects of players, or, in other words, the *mechanical player*.

Human-computer interaction (HCI) models divide a user into subsystems to evaluate processing bottlenecks and limitations – effectively viewing the player as a machine. This aligns more closely with our goal of modeling a mechanical player. Card, Moran, and Newell (1983) presented the Model Human Processor (MHP), which divides a generic user into three subsystems: motor, cognitive, and memory. Each subsystem has its own processor to handle tasks unique to that system. This division allows MHP to see which systems are the bottleneck for task completion. We adopt this

and fighting styles to describe the skills associated with that style that are required for competency.

approach, however we refined MHP's subsystems which were too abstract. To the best of our knowledge, the MHP viewpoint has not been applied to players. Here we detail the motor system as used to interact with games and analyze motor-focused challenges with respect to their associated motor abilities to find processing bottlenecks.

The Player Homunculus

We need to speak about players both specifically and generically. When we want to talk about a generic able-bodied neurotypical player, we will use the term *Player Homunculus*[2], an abstract representation of a player with normative motor and cognitive abilities.

Defining the Abilities of a Player Homunculus

We focus on the motor model in part because it is the simplest one to assess experimentally, but we cannot ignore all cognitive abilities. For the current homunculus iteration, we limit these to *perception, attention,* and *memory*, to align with the MHP model of users.

We investigated kinesiological models of muscle groups (Hamilton, Weimar, & Luttgens, 2011; VanPutte, Regan, & Russo, 2011), but found that individual muscles can belong to different groups when used to perform actions. We believe that this level of specificity would overly complicate our analysis, as different actions would nevertheless use the same muscle groups. For example, bending or rotating the wrist is used for both tracing a straight line or a circle with a stylus. Though the actions are different, they use the same muscles. Since actions are mediated by controllers, which are often offer limited means of interaction, we must distinguish between the actions performed by the player and the muscles used to perform them. Rather than muscle groups, we use controllers and actions possible through them.

We first look at controllers, focusing on standard controllers (e.g., Xbox

2. This is inspired by Penfield's Motor and Sensory Homunculus (Penfield & Rasmussen, 1950), which maps the relationship of information processing parts of the brain to various parts of the body.

One controller, Playstation 4 controller), handheld motion controllers (e.g., Wii Remote, Playstation Move), full body motion controllers (e.g. Kinect), smartphones, handheld consoles (e.g., Nintendo 3DS, Playstation Vita, Nintendo Switch), keyboards, mice, and fight sticks (arcade style controllers made for fighting games). We assume that the player is holding or interacting with them in the ergonomically intended manner. We list possible interactions (e.g., press button, pull trigger, shake controller) for each controller. We abstract from interactions to movement, so “pressing a button” becomes “pressing.” Table 2 gives the motor interaction for each controller type. We further classify these abilities into fine or gross motor abilities but indicate when an action could reasonably fall into both categories.

Table 2

Motor Abilities for Video Game Controllers Categorized as Fine Motor, Gross Motor, or Both

Category	Motor action	Hardware context
B	Pressing	Fight sticks, handheld consoles, handheld motion controllers, keyboards, mat controllers, standard controllers
F	Bumping	Handheld consoles, standard controllers
F	Pulling	Handheld motion controllers, standard controllers
B	Moving	Fight sticks, full body motion controller, handheld console, handheld motion controller, mice, standard controller
F	Swiping	Smartphones/tablets, standard controller
F	Pinch-to-zoom	Smartphones/tablets, standard controller
B	Swinging	Handheld motion controller
B	Pointing	Handheld motion controller
B	Shaking	Handheld consoles, handheld motion controller, smartphones/tablets
B	Drawing	Handheld consoles, handheld motion controller, smartphones/tablets
G	Thrusting	Handheld motion controller
F	Tilting	Handheld consoles, handheld motion controller, smartphones/tablets
B	Flicking	Handheld motion controller, smartphones/tablets
G	Positioning	Full body motion controller
F	Tapping	Handheld consoles, smartphones/tablets
F	Speaking	Handheld consoles, smartphones/tablets
F	Making facial expressions	Handheld consoles, smartphones/tablets
F	Clicking	Mice
F	Scrolling	Mice

Table 2: Motor interactions available for each controller type: F = Fine motor, G = Gross motor, B = Both.

Refining the Homunculus for games

We eliminate redundant actions from the list in Table 2, and refine abstract interactions to concrete motor abilities. We thus separate actions by the body parts used in the motion and their context. This lets us combine similar actions into specific groups. Combining the body part (e.g., finger) and action (e.g., pressing) into (e.g., finger pressing) to create motor abilities.

We show our refinement of fine motor abilities, to illustrate our reasoning.

As the full list is long, the details for the gross motor abilities have been omitted.

Fingers

One approximation is that pressing, clicking, tapping, pulling and bumping are the same action. *Pressing* is done by bending a finger or thumb at the knuckle to depress buttons on a controller; it exists in the context of standard and handheld motion controllers, handheld consoles, keyboards, and fight sticks. *Clicking* is done by bending a finger to depress the button on a mouse; this is the same as *pressing* as the orientation of the fingers and wrist is similar. As the physical actions are similar, over a similar time frame, we join them as the same action. *Tapping* is where players use their finger to touch a designated spot on a touchscreen; the motion used is identical to clicking and pressing, with an experiential difference due to the feedback difference between touchscreens and physical buttons. Consider a game like *Impossible Jump* (UltraRu, 2015) where the player must tap the screen to make their triangle avatar jump. Compare this action to that of *Bit.Trip Presents Runner 2* (Gaijin Games, 2013), where the player must press a button to make their avatar jump. The haptic feedback of the button press gives the player more subtle information about how quickly inputs can be registered. As we are concerned with isolating the motor abilities, we consider this difference as negligible and so group them together. *Pulling* is done by bending a finger to depress a trigger button; it exists in the context of standard and handheld motion controllers. The most common example is firing a gun in a shooter game like *Halo: Combat Evolved* (Bungie, 2001) where the right trigger is mapped to the gun's trigger. Like *clicking*, the only difference between *pulling* and *pressing* is the orientation of the player's hand, so we again group them together. *Bumping* is done by bending a finger to depress the shoulder button on standard controllers and handheld consoles. Example: shooting in *Final Fantasy VII: Dirge of Cerberus* (Square Enix, and Monolith Soft, 2006) that links the gun trigger to the R1 button. The player's hand orientation matches *pulling*, as does the description so we group *bumping* with the others. We encapsulate all these actions as *pressing*.

Similarly, swiping, flicking, and scrolling are the same action presented in different contexts. *Swiping* is when a player moves a finger or a stylus across an area of a touch-sensitive surface; it exists in the context of standard controllers, handheld consoles, and smartphones/tablets. *Flicking* is the interaction of quickly swiping across an area of a touch-sensitive surface; it exists on smartphones/tablets and handheld consoles. The difference between *flicking* and *swiping* is time; *flicking* is a rapid action, while *swiping* can be done at any pace. As we are looking to coarsely define actions, certain time differences are negligible. Therefore, we consider flicking and swiping to be the same (though an even finer model could separate them). *Scrolling* is where the player bends a finger to rotate a scroll wheel; it exists in the context of mice. We omit "scrolling" on touchscreens as it is really an adapted case of swiping. While *scrolling* involves finger bending motions like *pressing*, the mechanics differ. With *pressing* your finger is always moving inwards/towards your body, while *scrolling* moves both towards your body (moving the scroll wheel backward) and away from your body (moving the scroll wheel forward). *Swiping* similarly occurs both towards and away from the body depending on the direction of the swipe, and thus is closer to *scrolling*. *Swiping* and *scrolling* only differ in the choice of knuckle which bends; *swiping* motions tend to bend at the first knuckle (metacarpophalgeal joint, where the finger meets the hand), while *scrolling* tends to bend at the second or third knuckle. At our coarse level, there is no apparent effect on the time or experience of the motion due to this difference, so we group these actions together under the name *swiping*.

Pinch-to-zoom is the coordinated movement of two fingers to create a pincer-grip/pinching motion on a touch-sensitive surface. It exists in the contexts of smartphones/tablets and handheld consoles. It is independent from the other actions because of its motor coordination. This coordination can be measurably more difficult for different age groups; performing coordinated activities has been shown to increase cognitive load for older adults (Godde & Voelcker-Rehage, 2017; Lindenberger, Marsiske, & Baltes, 2000; Malcolm, Foxe, Butler, & De Sanctis, 2015; Papegaaij, Taube, Baudry, Otten, & Hortobagyi, 2014; Seidler, et al., 2010).

Single task coordinated actions (STCA), like *pinch-to-zoom*, differ from multi-task coordinated actions (MTCA). STCAs require movement coordination to accomplish a single specified goal, for example using thumbs to press tiles in *Piano Tiles 2* (Hu Wen Zeng, and Cheetah Games, 2019), where each thumb is responsible for part of the screen. MTCAs involve two non-coordinated single task actions at the same time, like controlling an avatar with the left thumbstick and the camera with the right. MTCAs affect cognitive load (and thereby perceived difficulty) of challenges but may not affect the motor difficulty. This is because MTCAs are asking players to simultaneously achieve two sub-goals, but motor difficulty is fixed with each interaction (i.e., pressing a button is always the same level of intrinsic difficulty).

Wrist/Forearms

At a first approximation, wrist movements are all the same except for speed requirements. The wrist is an ellipsoidal joint, offering a limited range of motion. Furthermore, players' wrist motions tend to be accompanied by forearm movement. Here we examine the differences between *pointing, flicking, tilting, drawing, swinging,* and *shaking.*

Pointing is the controlled movement of the wrist (mainly) used in the context of positioning a cursor using a handheld motion controller. With *pointing*, wrist movements are limited to lateral (wrist flexion and extension as Figure 2 shows, like waving as a greeting) and vertical (radial and ulnar deviation as Figure 3 shows, like fanning oneself) due to how the controller is held. Occasionally players may incorporate forearm movements to increase their range of motion.

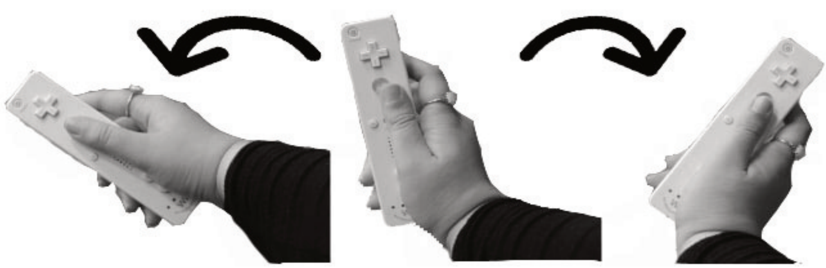


Figure 2: Lateral Wrist Pointing Movements. This figure illustrates wrist flexion and extension while holding a handheld motion controller.

Figure 3: Vertical Wrist Pointing Movements. This figure illustrates wrist deviation while holding a handheld motion controller.

In contrast to *pointing*, *flicking* is the quick lateral movement of the wrist used in the context of moving a cursor from one position to another with a handheld motion controller. However, *flicking* is discrete while *pointing* is a continuous action; this difference affects completion speeds and how/where these actions appear in a game. *Pointing* exists in accuracy tasks (e.g., archery *Wii Sports Resort* (Nintendo EAD Group No. 2, 2009)) and can be a challenge on its own. *Pointing* can also appear alongside pressing actions. *Flicking* exists as a supporting motion in many challenges. For example, serving the ball in table tennis for *Wii Sports Resort* (Nintendo EAD Group No. 2, 2009). As it is less accurate than *pointing*, *flicking* appears less frequently. Even though the underlying wrist movements are the same, this difference in game contexts merits keeping them separate.

Tilting involves moving entire controllers using coordinated wrist and forearm movements; it exists in the context of handheld motion controllers, smartphones/tablets, and handheld consoles. The way that each device is held affords different degrees of movement. For handheld motion controllers when held in a single hand, tilting laterally involves the player twisting their wrist and forearm to angle their controller in the same motion as turning a doorknob (wrist supination and pronation Figure 4). Tilting vertically in this context is the same movement as vertical pointing movements (radial and ulnar deviation). For smartphones/tablets held in a single hand in portrait mode, tilting is the same as handheld motion controllers.

Figure 4: Lateral Wrist Tilting Movements Single Handed. This figure illustrates wrist supination and pronation while holding a handheld motion controller.

When holding devices between the hands (as in landscape orientation), tilting vertically uses the same movements as vertical pointing (radial and ulnar deviation). When tilting laterally, the wrist's main function is stability. The tilting motion is a coordinated movement of the forearms (forearm flexion and extension Figure 5). For example, when holding the Nintendo Wii U gamepad, tilting the device laterally to the left requires the player's right forearm to move up (flexion), while their left forearm simultaneously moves downward (extension). The player's wrists remain stable in order to not drop the controller. An example is steering the flying beetle item in *The Legend of Zelda: Skyward Sword* (Nintendo EAD, 2011) by tilting the handheld motion controller. Tilting is a continuous action, like pointing, but the added twisting movement is enough difference to keep them separate.

Figure 5: Lateral Wrist Tilting Movements Two Handed. This figure illustrates forearm flexion and extension while holding a handheld console (Wii U).

Drawing is the interaction of moving a brush proxy in a controlled path, over a canvas, using predominantly wrist and forearm movements. It exists in the context of handheld motion controllers (held in a single hand), smartphones/tablets, and handheld consoles, where the brush proxy is either a finger or a stylus used to paint on a touchscreen (canvas). *Drawing* motions depend on the scale of the canvas, with larger canvases using more forearm movements, and smaller ones using more wrist movements. This scale dependency makes *drawing* distinctly different from the previously discussed actions.

Swinging is the repeated lateral movement of the wrist in the context of handheld motion controllers held in a single hand. Examples include using tools like the fishing rod and net in *Animal Crossing: City Folk* (Nintendo EAD Group No. 2, 2008), cracking an egg in *Cooking Mama: Cook Off* (Cooking Mama Ltd., 2007), and sword actions in *The Legend of Zelda: Skyward Sword* (Nintendo EAD, 2011). To be considered *swinging*, a minimum of two distinct lateral wrist movement must occur (back and forth), though more can be performed to repeat the in-game actions; in contract flicking is a single movement. The difference between *swinging* and *flicking* is speed; flicking is fast and less precise, while swinging can be steady and accurate.

Shaking is the quick repetitive movements of the wrist and/or forearm to move a controller. It exists in the context of handheld motion controllers (both orientations), smartphones/tablets, standard controllers, and handheld consoles. For handheld motion controllers held in one hand and smartphones/tablets in portrait mode, *shaking* exists as either a vertical wrist motion (radial and ulnar deviation) mimicking the motion of a drumstick tapping on a drum, or as a jerking forearm movement similar to the motion of shaking a cocktail shaker as Figure 6 illustrates. Examples include ground pound in *Donkey Kong Country Returns* (Retro Studios, 2010) when using a handheld motion controller and asteroid in *SpaceTeam* (Sleeping Beast Games, 2012) on smartphones/tablets. For handheld motion controllers held horizontally, smartphones/tablets in landscape mode, standard controllers, and handheld consoles (which are held between the hands), *shaking* is exclusively the result of forearm movement (forearm flexion and extension). Though shaking actions are possible for all these controllers in this orientation, they are most common for handheld motion controllers. Examples include: ground pound in *Donkey Kong Country: Tropical Freeze* (Retro Studios, 2014), performing wheelies in *Mario Kart 8* (Nintendo EAD, 2014) and performing the homing hat throw in *Super Mario Odyssey* (Nintendo EAD, 2017).

Figure 6: Forearm Shaking Movements. This figure illustrates forearm rotation while holding a handheld motion controller.

We were unable to find examples of *shaking* for landscape smartphones/ tablets and handheld consoles. We conjecture that because the screen is attached, shaking the controls shakes the screen too, making the game difficult to play since the player can't receive visual feedback easily. The movements for all *shaking* contexts are sufficiently distinct from all previously discussed wrist movements.

Neck and face

Head movements, such as *tilting*, *nodding*, and *shaking* are neck movements. These actions are becoming more important for AR and VR games, which use headsets and monitor head movements as input. But these are out of our current scope.

A face's actions are making facial expressions and speaking. *Facial expressions* are registered by the front camera of handheld consoles (e.g. *Pokémon Amie Pokémon X and Y* (Game Freak, 2013)). *Speaking*, as an action, exists for smartphones/tablets and handheld consoles and is performed by making noise directed at the device's microphone. *Speaking*, as we describe it here, is not to be confused with natural language processing. The microphones are only detecting whether a noise is made and at what intensity. Examples include Puzzle 138 in *Professor Layton and the Diabolical Box* (Level-5, 2009), which requires players to blow into their microphone simulating a gust of wind, and *Chicken Scream* (Perfect Tap Games, 2017) on smartphones, which allows the user to control how the chicken avatar moves by making sounds.

Ankle and feet

Existing controllers that use foot input (mat controllers) only allow for pressing as an action. Therefore, even though there are many potential movements for ankles and feet, we are limited to considering the two as a single unit and to condense all possible actions to just *pressing*. Examples include *Dance Dance Revolution* (Konami, 1999), *Shaun White Skateboarding* (Ubisoft Montreal, 2010), and *Mario and Sonic at the Winter Olympic Games* for the Wii Balance Board (Sega Sports R&D and Racjin, 2009).

The Generic Player Homunculus.

A generic player homunculus has motor abilities and basic cognitive abilities (attention, perception, memory). For each ability, a player has a score between 0, meaning not able to use that ability, and 100, meaning fully able to use that ability unencumbered. The generic player homunculus being an abstract representation of an able-bodied neurotypical player can use all abilities to their fullest.

Table 3 includes the refined set of motor abilities.

Table 3

Fine and Gross Motor Abilities Used to Interact with Video Games

Type of motor movement	Body part category	Specific body part	Action
Fine motor abilities	Hands	Fingers	Pressing
			Swiping
			Pinching
		Wrist	Shaking
			Flicking
			Pointing
			Swinging
			Drawing
	Head	Neck	Tilting
			Moving
		Face	Speaking
			Making facial expressions
	Feet	Ankle and foot	Pressing
Gross motor abilities	Arms		Pushing
			Swinging
			Drawing
			Rotating
			Positioning
	Legs		Moving
			Positioning
	Torso		Positioning

The generic player homunculus is not the representation of all players. Player homunculi need to be constructed for different demographics in order to more accurately represent their abilities. This concept can be used to describe groups of players as well as the very specific skill set of a single player. Most importantly, homunculi can be constructed experimentally, or approximated through the literature of various fields.

GAMEPLAY CHALLENGES

We define a *gameplay challenge* as any in-game activity with a success condition which engages the player in a way that requires some level of proficiency in at least one dimension (physical or cognitive). One view of games is that they can be described adequately by the set of challenges they use (Adams, 2010; Djaouti et al., 2008; Feil & Scattergood, 2005;

McMahon et al., 2015; Veli-Matti, 2014). This positions *challenges* as the unit tasks of gameplay, describing individual tasks the player must accomplish. This reductionist approach does not extend to all aspects of the player experience, but is sufficient to study the mechanical experience. Though previous works (e.g. Adams, 2010; Feil & Scattergood, 2005; Veli-Matti, 2014) differ on what makes a challenge, they all define them along the lines of goals and mechanics.

We look to identify atomic types of challenges – challenges that are mutually exclusive in their goals, context, and mechanical experience. Others have tried to produce such lists (e.g. Adams, 2010; Djaouti et al., 2008; Feil & Scattergood, 2005; McMahon et al., 2015). To decide if we can use one of these lists, we need criteria to judge their challenge descriptions. Ideally, a challenge description should include the in-game mechanics and the mechanism of interaction between the player and the game (i.e. the inputs and outputs). The mechanics will let us understand the goals and actions of the challenge. The mechanisms of interaction provide the mechanical context and some insight into the mechanical experience. We found six frameworks for analysing gameplay and categorizing challenges (Adams, 2010; Bjork & Holopainen, 2004; Djaouti et al., 2008; Feil & Scattergood, 2005; McMahon et al., 2015; Veli-Matti, 2014). All frameworks covered the in-game mechanics; none included the mechanisms of interaction.

We therefore needed to either refine or create a new taxonomy that fit our purposes. To direct this process, we needed to decide the components of an ideal challenge description. The purpose of the refinement is to arrive at a list of atomic challenges. Therefore, just adding mechanisms of interaction is insufficient as that only gives us *insight* into the mechanical experience. We need to understand the specific abilities underlying the mechanisms of interaction to know the mechanical experience. To this end we explored the concept of competency profiles.

As unit tasks, we can characterize challenges by their *competency profiles*, the set of human abilities (motor, cognitive, emotional, etc.) needed to succeed at the task (Fleishman et al., 1984). This is different from game mechanics, which are described in game terms (e.g., match three blocks in a line). Competency profiles deal with specific task-based abilities (i.e., pressing a button with your finger). By characterizing challenges through their competency profiles, we can get an idea of the intrinsic mechanical experience of each challenge. Evaluating this mechanical experience requires approximating the weights of the abilities in the competency profile – creating an *intrinsic competency profile*. The *intrinsic competency profile* informs the designer about the *expected mechanical experience* of the challenge for able-bodied, unencumbered players.

A good challenge description must delineate between similar challenges, and thus include the following:

1. the in-game mechanics associated to the challenge
2. the mechanism of interaction between the player and the game
3. the intrinsic competency profile

The first and second point lets us capture the challenge's goals and context, while the third point is what lets us to distinguish between similar challenges with different player mechanics.

Perspectives on challenges

No list was perfect, so we decided to use Adams (2010) as a starting point. We use Adams' list because of its grounding in gameplay examples and understanding of challenges being physical and cognitive. He presents 10 major challenge types, subdivided into 30 challenges (Adams, 2010, p. 19). This list attempts to capture both the in-game mechanics and the player experience. The major challenge types give an idea about the expected mechanical experience and whether it focuses on cognitive or motor abilities, while the individual challenges provide more insight into the particular mechanics for that challenge. Consider the *Timing and Rhythm* challenge. Adams (2010) defines rhythm challenges as, "tests of the player's ability to press the right button at the right time" (p. 263) directly referencing the mechanism of interaction and the mechanic. From this definition we can tell that these challenges emphasize motor abilities, justifying it falling under the *Physical Coordination* type. However, it does not provide crisp definitions for each category, leading to different experiences being lumped together.

Refinement methodology

Adams' (2010) definitions are inconsistent, with varying type of information in each. We refine the definitions to systematically describe game mechanics, control mechanisms, and content of the challenge (e.g. single vs. multi-player, time limits, etc.). Often when the definition was not specific, we could synthesize the information from the examples provided. This required playing the games involved, watching other players interact with the game, and attempting to list out the traits of the games involved to look for similarities.

Once we had consistent definition and examples, we found other gameplay instances that fit those descriptions. We tried to find as many as possible across various "genres" and systems. The purpose of these examples was not to determine whether these challenges are universal, but rather to get a better understanding of where they tend to appear and how they exist. This process relied on the subjective knowledge of the researcher (and, to a limited extent, lab peers) to come up with examples, as extant literature on challenges from this perspective is limited, there was no easy way to systematically search for this information. Our collective gaming experience spans more than 20 years, covering the third to eighth generations of home consoles, arcade games, and home computers from MS-DOS to Windows 10. We average 15-20 hours of gaming per week between a variety of game genres (MMOFPS, Hack 'n' Slash, Puzzle, Strategy, Fighting, and casual). This a limitation of the process; future work would benefit from a larger pool of researchers with different gameplaying experiences.

We sorted our examples by their *mechanisms of interaction,* as this is the most easily identified difference. We did this by examining the game mechanics and instructions for each instance and the controller used. This first separation accounted for differences in motor abilities used even if the abstract goals are the same. For example, playing the guitar in *Rock Band* (Harmonix, 2007), *Donkey Konga* (Namco, 2004), and *Just Dance* (Ubisoft Paris and Ubisoft Milan, 2009) all use *Timing and Rhythm* challenges. The mechanics require you to stay in time with the song and react to the on-screen stimuli, but this is accomplished in broadly different ways (pressing buttons, hitting a drum, and swinging your arms). We used *close reading techniques* to construct the competency profiles, first identifying required motor abilities (see previous section for list of abilities) then comparing with the generic player homunculi in order to find the normative mechanical experience.

For each category, we added more examples that fit the specific description of the challenge. We separated instances based on game mechanic variants, capturing distinctions like pushing one or two buttons, or having time limits on the challenge. Our purpose was to capture differences in experience due to increased use of attention for coordinated movements or the use of perception. This allowed us to understand the broad cognitive abilities of the competency profile. We then ranked the competency profile abilities as: not used, used but not noticeably, noticeably used, important but not limiting, or limiting ability.

We repeated this process (added more examples, found additional distinctions, etc.) until it stabilized. We did not have to split abilities more than twice. We then examined the rest of the context in which these challenges occur: whether the game is competitive or cooperative, single or multiplayer, team-based or solo, etc. The purpose was to see whether differences in context creates differences in the motor or cognitive abilities used, leading to further refinement when that was the case.

We considered two challenge instances to be identical if they involved the same motor and cognitive skills from a player, occurred over similar periods of time, and were performance bounded by the same skill. We then re-examined our observations of the examples to assign values to each ability in the competency profile. We assigned each ability a value between 0 and 100 with a margin of error of (at least) ±10 as an indication of “percentage of use”; this helped us understand the relationship between abilities in the same category or on the borders. For example, a value of 37 for a skill S would correspond to “a player uses 37% of their (normative) skill S while completing this challenge.” Of course, 37% is ridiculously over-precise: we only distinguished the five categories outlined above: not used, used but not noticeably, noticeably used, important but not limiting, or limiting ability. The use of a “finer scale” is to allow for increased precision in the future, we expect that “important but not limiting” will warrant refinement. The extra range also allows close readers to express their feeling of finer differences in the use of each ability. While this is still subjective, we sample-tested our assignments against others' subjective classification (within our lab) and found our rough numbers to be uncontroversial. Currently our descriptions only concern an individual's mechanical experience of these challenges. Thus, while we included examples of multi-player games, we examined them when playing with or against humans or non-player characters.

More specifically, when doing a close reading of a challenge, we played the game several times, first to become familiar with the challenge, then to witness our own use of each skill in the performance. We systematically observed our performance and ranked our use of each ability as it related to the completion of the challenge. A first pass established the gross scale (the five categories), and then subsequent passes refined that into a number that expresses a subjective value judgement of relative use as compared to other like challenges.

These values, like the challenges themselves, are a starting point and we will experimentally validate them in the future.

We applied this process to Adams' *Speed and Reaction Time* challenges, which we previously noted as incorrectly lumped together. We split them and analyzed the *Speed Challenges* in detail. This illustrates that refining the definitions, via our 3-pronged approach, leads to new distinct categories, each with *simpler* descriptions. Refining Adams' complete list of challenges is still work in progress.

Speed Challenges

Speed challenges "test the player's ability to make rapid inputs on the controls" (Adams, 2010, p. 262). Thus, these challenges have a *time limit*; otherwise the idea of *rapid* wouldn't be well defined. Furthermore, these challenges should be identifiable as small chunks of gameplay, not something that takes place over the course of hours of a play session. Secondly, "inputs on the controls" indicates that this ought to be controller independent, and so examples should exist using all controller types.

Finally, Adams implies these are motor-focused challenges by placing them under *Physical Coordination*. This definition does not mention stimulus that would trigger this action. This is likely due to the distinction between *Speed* challenges and *Reaction Time* challenges, where the latter relies on a specific stimulus for a reaction. So, gameplay instances that require players to "react" and not just "act" do not belong to *Speed Challenges*. We see *short sessions, time limits, and exclusively motor-focused* as the defining features of this challenge type.

Adams lists *Tetris* (Pajitnov & AcademySoft, 1986), *Track & Field* (Konami, 1983), and *Quake* (id Software, 1996) as examples, without giving specific instances inside these games to pinpoint what he means. He does list platformers, shooters, and fast puzzle games as genres where these are most readily found. Deeper analysis reveals more instances of reaction time over speed challenges. We identified several examples of gameplay instances that had *short sessions, time limits, and are motor-focused*. We started our survey with party games and games that relied on mini-games, as they are explicitly designed as short session challenges with time limits. Nintendo games are particularly popular in this genre and exist across multiple input mechanisms; this gave us 10 examples:

- Manic Mallets, *Mario Party 5* (Hudson Soft, 2002)
- Cycling, *Mario and Sonic at the Olympic Games* (Sega Sports R&D, 2008)
- Mecha-Marathon, *Mario Party 2*(Hudson Soft, 2000)
- Pedal Power, *Mario Party* (Hudson Soft, 1999)
- Tenderize the Meat, *Cooking Mama* (Cooking Mama Ltd., 2006)
- Impressionism, *Wario Ware: Touched!* (Intelligent Systems & Nintendo SPD, 2005)
- Wash Rice, *Cooking Mama* (Cooking Mama Ltd., 2006)
- Hammer Throw, *Mario and Sonic at the Rio 2016 Olympic Games* (Sega Sports R&D & Racjin, 2011; Sega Sports R&D & Racjin, 2012)
- Candy Shakedown, *Super Mario Party* (NDcube, 2018)

- Trike Harder, *Super Mario Party* (NDcube, 2018)

These gameplay instances are rather different in their mechanisms of interaction, and thus the motor ability that each emphasizes. Analyzing the description of each challenge gave rise to the following new sub-categories of *Speed* challenges: *button mashing, rapid analog stick rotation, rapid tapping, scribbling, rapid controller rotation, and rapid controller shaking*. For space considerations we only present the decomposition of button mashing. Readers interested in viewing the results of the other speed challenges can contact the first author directly.

Button Mashing

Button Mashing is where a player must rapidly press button(s) or key(s) in a given time limit. While button mashing retains short play sessions, time limits, and motor focus, it becomes hardware-dependent in that these challenges require real, physical controls to depress (ergo "buttons" to "mash"). This is different than pressing virtual buttons, like those found on a touch screen, as it loses the mechanical feedback of a button. From our list, button mashing appears in:

- Manic Mallets, *Mario Party 5*
- Mecha-Marathon, *Mario Party 2*
- *Track & Field*

We can then easily find more (Nintendo) instances:

- Psychic Safari, *Mario Party 2* (Hudson Soft, 2000)
- Speed Skating, *Mario and Sonic at the Winter Olympic Games* (Sega Sports R&D & Racjin, 2009)
- Ridiculous Relay, *Mario Party 3* (Hudson Soft, 2001)
- Take a Breather, *Mario Party 4* (Hudson Soft, 2002)

- Pump, Pump, and Away, *Mario Party 3* (Hudson Soft, 2001)
- Chin Up Champ, *Wii Party* (Nintendo SPD Group No. 4 & ND Cube, 2010)
- Balloon Burst, *Mario Party* (Hudson Soft, 1999) and *Mario Party 2* (Hudson Soft, 2000)

The abundance of examples argues that this is a common category of challenge in the party and mini-game genres. Identifying examples outside Nintendo and party games is made more difficult as instances tend to be embedded in larger gameplay segments. Here are four more representative examples:

- Torture Attacks, *Bayonetta* (Platinum Games, 2010) and *Bayonetta 2* (Platinum Games, 2014)
- Dragon's Breath, *South Park: The Stick of Truth* (Obsidian Entertainment & South Park Digital Studios, 2014)
- Boss Knockouts, *Donkey Kong Country: Tropical Freeze* (Retro Studios, 2014)
- Colossus of Rhodes Fight, *God of War 2* (Sony Computer Entertainment Santa Monica Studio, 2007)

The time limit is now implicit, often being tied to the length of an animation or just not explicitly shown to the player. The previous examples all had explicit timers or gauges. Nevertheless, we didn't find that explicit versus implicit time limits affected our mechanical experience. Generally, we were too focused on pressing quickly to watch the timer when it was explicit. As well, since the goal in every instance is to press the buttons as quickly as possible, there was no change in our play style or strategy. This is likely because of the simplicity of this particular challenge; we believe explicit time limits would affect cognitive-focused challenges more.

Having 16 examples for button mashing shows it's an easily identifiable and common challenge. But do these examples have the same game mechanics? Consider three of our original examples: Manic Mallets, Speed Skating, and Mecha-Marathon. In Manic Mallets, the player hits a single button as many times as possible in the time limit; Speed Skating requires the player alternate between two buttons; Mecha-Marathon requires the player to press two buttons simultaneously as many times as possible. Manic mallets, with its single button, is a straightforward case of button mashing, requiring no additional abilities outside of pressing the button. Mecha-Marathon requires some coordination of button pressing, principally focussing on the pressing but requiring some attention. Speed Skating similarly requires finger pressing and attention, adding a small perception and memory component to keep the alternating pattern correct. All the other examples repeat one of these three patterns. These differences identified in used abilities yielded distinctions in button mashing based on type of input: single, multiple, and alternating.

Single Input

Single Input Button Mashing tasks the player with repeatedly pressing a specific single button or key on the controller as fast as possible within a given time limit. Examples include: Manic Mallets (*Mario Party 5*), Dragon's Breath (*South Park: The Stick of Truth*), torture attacks (*Bayonetta and Bayonetta 2*), and Boss Knockouts (*Donkey Kong Country: Tropical Freeze*). All have the same goals, mechanics, and mechanisms of interaction. We can now examine their competency profiles based on close readings of these instances.

In Manic Mallets two players repeatedly hit a switch with a hammer to avoid being crushed by a bigger hammer. The time limit is explicit at ten seconds, and the team with the highest score wins. The context is local team-based cooperative-competitive multiplayer on a standard controller. We played this game in multiple scenarios with both normal difficulty NPCs and human players to determine the differences between modes of play. Our first few playthroughs were with an NPC partner versus NPC opponents. The impression from these were that the partners hits weren't reliable for winning, and so we focused entirely on our own button presses. Here *rapid finger pressing* using our forefinger was the most important movement. We never watched the opposing team and were solely concerned with ourselves. We relied minimally on attention to make sure we were pressing the right button at a good pace. Perception's effect also seemed minimal after the initial press on the right button. Playing games with a human partner against normal difficulty NPCs, we (implicitly) realized that we could rely on our partner, but that our mechanical interactions were the same: finger pressing and minimal attention and perception. The main difference was our impression of urgency; knowing we had a partner whose contribution was meaningful put less pressure on us to perform optimally. This change in our feeling of pressure did not affect the competency profile as it didn't change our approach to the gameplay. When we played with a human partner against a human team, we limited ourselves to holding the controller as intended but did not limit our teammate or opponents in the same way. The mechanical interactions were again the same, but the performance pressure was significantly higher; we pushed the buttons much harder, leading to fatigue over repeated plays. Removing the limitation of ergonomic holds may have resulted in a different ability emerging in the competency profile to adjust to this pressure. Overall, we identified *finger pressing, attention, and perception* as the three abilities in this competency profile.

Dragon's Breath (Figure 7) is a mage-class attack where the player repeatedly waves a lit firecracker in their opponent's face to deal damage. The time limit is implicitly tied to the length of the waving animation, with every button press adding to the base damage of the attack. As this exists as a single attack in a larger combat system, failure to perform doesn't guarantee loss, but does hurt the player's ability to play optimally. The context is local character-based competitive single player on a keyboard and mouse. Unlike Manic Mallets, there was only one scenario to explore, which seemed the same as Manic Mallets when we were playing exclusively with NPCs. Finger pressing is the most important ability, with minimal attention and perception after the start. Since there was no immediate threat of loss, we didn't feel intense pressure to continuously perform optimally or extend ourselves beyond a comfortable level. Therefore, this example and Manic Mallets have the same competency profile.

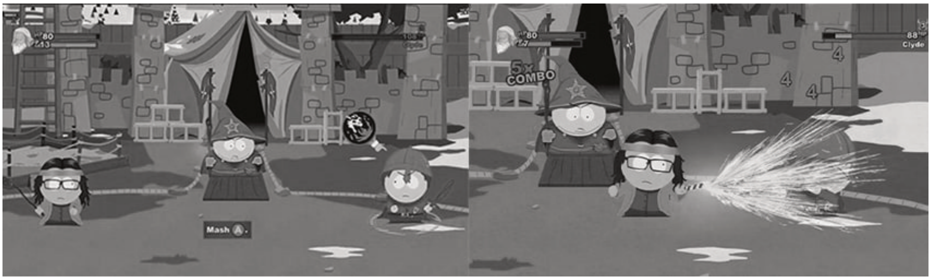


Figure 7: Dragon's Breath Gameplay. This figure illustrates an example of single input button mashing with implicit time limit in the form of South Park: The Stick of Truth's Dragon's Breath attack.

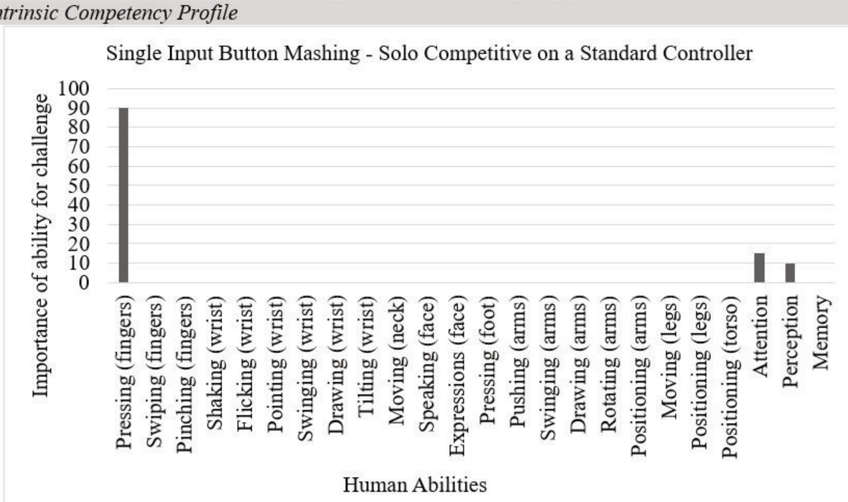
Torture attacks are a triggered combat action which removes the player from regular combat to perform a quick button mashing segment to increase their score and deliver a cinematic finishing blow. Like Dragon's Breath, the time limit is implicit and tied to the length of the animation. The context is local solo competitive single player on a standard controller with only one scenario. This segment of combat seemed mechanically identical to the previous ones: finger pressing, minimal attention, and perception. Though there was no immediate fear of failure from not getting the highest damage, the knowledge of rewards for high scores after the mission created some pressure to perform well, but did not change the competency profile.

Boss Knockouts are cinematic finishers to a boss fight where the player mashes a button to increase their high score. The time limit is implicit. The context is a local solo competitive single player on a horizontal handheld motion controller. Like Dragon's Breath, there is no risk of failure, and so little pressure to perform optimally. The competency profile is thus the same.

These instances having the same competency profiles mean that we don't need to decompose this challenge further. Regarding the abilities used: In all examples, finger pressing speed seemed to be the main bottleneck, especially as pressure increased. Continuous plays of Manic Mallets made this clear, where the physical fatigue and slower pressing affected the outcome. Even in lower pressure scenarios, like Dragon's Breath, finger fatigue was the difference between defeating an enemy in one attack or needing more. We approximated the level of use of finger pressing to be 90, making it the limiting ability. We used a value of 90 to communicate that we're confident that testing will reveal as the limiting ability, but conscious of potential error. We found that attention and perception are both used, but not noticeably enough to imperil success. From play, the importance of these abilities was directly related to perceived pressure; attention increased when playing against human opponents. Nevertheless, we could still be competitive while holding a conversation with our opponents and/or teammates. We assigned attention a 15: used, but not important. With the margin of error of our estimates, we allowed that in sufficiently stressful circumstances attention may cross over into noticeably used territory. Perception seemed to have minimal effect on our performance. Beyond awareness that we were pressing the right button, we could play these instances blindfolded if given a cue to start. Thus, we assigned perception a 10, leaving it squarely in the used, but not important category, even with the margin of error. We summarize the intrinsic competency profile in Table 4.

Table 4

Challenge Description for Single Input Button Mashing Challenges

Name	Single Input Button Mashing
Challenge Information	
Definition	Repeatedly pressing a specific single button or key on the controller as fast as possible within a given time limit.
Mechanics	
Button pressing, short time limits	
Variable components	• Time limit
Context	
Mechanism of Interaction	Pressing buttons
Controller Type	Standard Controller
Number of Players	Single player
Type of Play	Competitive (solo)
Examples	
• Dragon's Breath, South Park: The Stick of Truth • Torture Attacks, Bayonetta and Bayonetta 2 • Boss Knockouts, Donkey Kong Country: Tropical Freeze	
Variants	
• Controller Type: Handheld console, handheld motion controller (horizontal), keyboard • Number of Players: Multiplayer • Type of Play: Cooperative-Competitive (team based), Cooperative	
Intrinsic Competency Profile	
	
Source of Difficulty	Finger pressing

Multiple Input

Multiple Input Button Mashing (MIBM) requires the player to push multiple buttons simultaneously, repeatedly, and rapidly. Examples include Mecha-

Marathon (*Mario Party 2*), and Chin-Up Champ (*Wii Party*). All examples we found have the same goals, mechanics, and mechanisms of interaction and thus does not need further decomposition. We analyzed these examples for their competency profile.

In Mecha-Marathon (Figure 8), each player competes against the others to wind up a doll by simultaneously mashing the A and B button (on a standard controller) within a 10-second time window, after which the dolls begin to fly forward. The doll that travels furthest wins. We played with normal level NPCs and human players to test whether the type of competitor affected the competency profile. Against NPCs we found ourselves pressing the two buttons with our forefinger and middle finger while bracing the controller body against our thigh. Performance was adequate, but the position was uncomfortable and repeated play was fatiguing. But this play style revealed that *rapid finger pressing* was the most important movement. We also actively noticed our wrist needing to be stable to allow for quick presses; this is the *wrist pointing* motion. We noticed that our perception wasn't actively used outside of understanding which buttons to push. Our attention seemed divided in this case, as we coordinated the simultaneous button pressing. In a second attempt, we found a more comfortable position where we held the controller in one hand, rested our thumb across the A and B buttons, with the pressing action being done as a movement of the base knuckle of the thumb (near the palm). However, it didn't improve performance against the NPCs and it required a bit of attention to make sure our thumb didn't slip out of place instead of to coordinate movement, but otherwise did not change other aspects of the profile. Against human players, in both holding contexts, we noticed that we exerted ourselves more as we actively considered the competition; we noticed increased movement of our wrist and forearm. The motion speed came from shaking our forearm. We replayed the NPC context to see whether this motion was used there without us noticing, and found that we were subtly moving our forearms, leading us to believe this action is important as difficulty increases.

Figure 8: Mecha-Marathon Gameplay. This figure illustrates an example of multiple input button mashing in the form of Mario Party 2's Mecha-Marathon.

Chin-Up Champ has players compete at performing the most chin-ups in 10 seconds by simultaneously mashing the A and B buttons on the Wii Remote held vertically. The context is local solo competitive multiplayer on a vertical handheld motion controller. We played against normal NPCs and human players. To play the game we held the remote in our right hand with our thumb on the A button and forefinger on the B button. The gameplay for both contexts was identical to Mecha-Marathon; emphasis on finger pressing, noticeable attention use, wrist stability through pointing, and minimal perception. We did not experience forearm shaking. We think this was predominantly because of the shape of the controller; the placement of the A and B buttons on the Wii Remote was more ergonomic resulting in a more natural hold and movement in comparison to the N64 controller as Figure 9 highlights.

In general, standard controllers assume that the player's thumb is their main interaction with the face buttons. This limits comfortable ways to hold the controller, leading to using other abilities to compensate for an uncomfortable grip. This seems to indicate that the specific motor ability that limits success not only changes with difficulty, but also with the controller. Thus, controller design, and choice of which buttons to press, causes variation in the competency profile.

In our survey we couldn't find many examples MIBM. While the mechanic of pressing two buttons simultaneously is used in other challenges, like quick time events, it doesn't frequently occur in a Speed Challenge setting. We conjecture that MIBM is less popular because of the difficulty in coordinating multiple simultaneous button presses. It can also explain why three button input is not used, as it would be too taxing on the player's cognitive and psycho-motor skills. The effect of controller variability on difficulty must also play into this, as designers may intuitively feel the discomfort of using this challenge in most controller contexts that expect thumbs pressing the face buttons. Another potential reason for the unpopularity of MIBM is the similarity in skills used in the single input button mashing. Designers may not consider them to be different enough and thus choose to use the cognitively simpler single input button mashing instead.

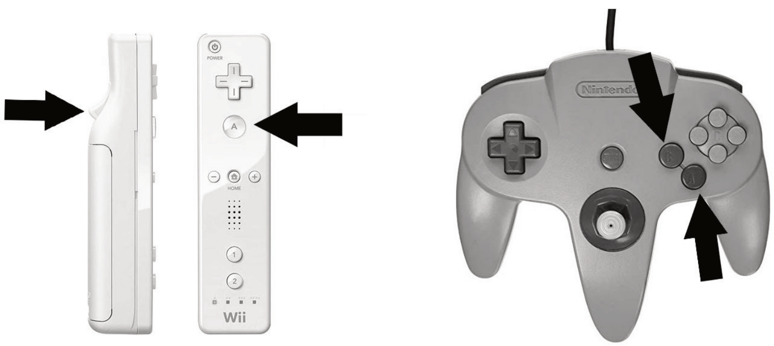


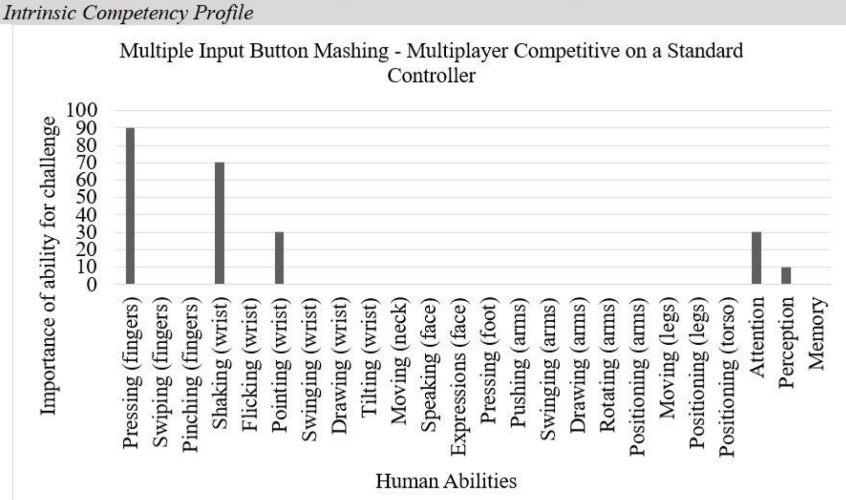
Figure 9: Comparison of A and B button placements between Wii Remote and N64 controller. This figure illustrates the different placement of buttons on the Wii Remote and N64 Controller.

Table 5 shows the competency profile, although the paucity of examples available means there is more room for error. Unlike SIBM, the MIBM competency profile does vary depending on the example. In both examples—Mech-Marathon and Chin Up Champ—finger pressing speed seemed to be important; whether it was using different fingers to press the two buttons or one finger to press both, this action was instrumental in the action. This was most obvious in comparing the different holds of Mecha-Marathon against NPCS and how there was no difference between our performance in these contexts. While high pressure situations introduced a new ability, finger pressing was the most essential ability. We approximated the level of use of finger pressing to be 90, making it the limiting ability. Wrist pointing is a supporting ability, as it was needed to stabilize the controller; it was never taxing, so we estimate it as noticeably used (30). Attention is like wrist pointing, noticeably used (30) in all scenarios, although for different reasons. Perception, on the other hand, had minimal effect on our performance, thus we assigned it a 10 (used, not important, and less so that the others). Since wrist/forearm shaking only appeared in a single variant (high pressure standard controller contexts), it was not a limiting ability of the abstract challenge type. However, because significantly affected our performance against human players in Mecha-Marathon, we gave it a 70, as important but not limiting. Given our built-in margin of error, this documented the importance of this ability, while leaving room for it to cross into the territory of limiting ability as difficulty rises.

Table 5

Challenge Description for Multiple Input Button Mashing

Name	Multiple Input Button Mashing
Challenge Information	
Definition	Pushing multiple buttons simultaneously, repeatedly, and rapidly.
Mechanics	
Button pressing, short time limits	
Variable components	• Number of buttons pressed • Which buttons are being pressed • Time limit
Context	
Mechanism of Interaction	Pressing buttons
Controller Type	Standard Controller
Number of Players	Multiplayer
Type of Play	Competitive (solo)
Examples	
• Mecha-Marathon, Mario Party 2	
Variants	
• Controller Type: Handheld console, handheld motion controller (vertical), handheld motion controller (horizontal), keyboard • Number of Players: Single player • Type of Play: Cooperative-Competitive (team based), Cooperative	
Intrinsic Competency Profile	



Source of Difficulty	Finger pressing

Alternating Input

Alternating input button mashing requires players to repeatedly and rapidly press two specific buttons in sequence. Examples include Psychic Safari (*Mario Party 2*), Take A Breather (*Mario Party 4*), Pump Pump and Away (*Mario Party 3*), Balloon Burst (*Mario Party and Mario Party 2*), Ridiculous Relay (*Mario Party 3*), Speed Skating (*Mario and Sonic DS*), and the Colossus of Rhodes fight (*God of War 2*). Adams included the Track and Field example here, but for brevity we will skip it. We analyzed the competency profiles of this group which has converging mechanics, goals, and mechanisms of interaction.

Psychic Safari (Figure 10) tasks two players to power up an ancient relic to destroy their opponent's relic. There is an explicit five-second time limit and the player who can make the most inputs wins. The context of this game is local solo competitive multiplayer on a standard controller. We played this game against a normal level NPC and a human opponent. We noticed a similar holding issue to Mecha-Marathon, as they use the same controller. We resorted to holding the controller with one hand while pressing the buttons with our forefinger and middle finger. In both contexts (NPC and human), we relied predominantly on finger pressing to work the buttons, with wrist pointing acting as a supporting ability. Our attention was used to keep the alternating pattern going and perception was used to know which buttons to press. Memory was trivially used, as the sequence needed was short enough to fit in short term memory.

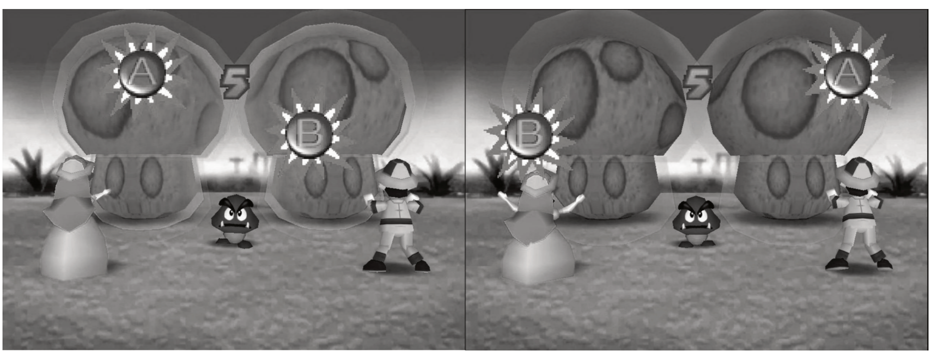


Figure 10: Psychic Safari Gameplay. This figure illustrates an example of alternating input button mashing in the form of Mario Party 2's Psychic Safari.

Take a Breather gets players to inhale by alternately mashing the L and R buttons to see who can hold their breath underwater longest. These is an explicit time limit of five seconds, after which the players submerge and the person who made the most inputs wins. The context is local solo competitive multiplayer on a standard controller. We played this against normal NPCs and human players. In both contexts we found the same abilities as Psychic Safari. The main difference between Psychic Safari, and this, was the ergonomics of play; by having the player press the shoulder buttons (L and R) we were able to hold the controller in a natural way. We still needed wrist pointing to provide stability when holding the controller and enacting the pressing, but now pressing was coordinated across both hands. We do not believe this affected the amount of attentional resources needed, as we were not coordinating our movements to be simultaneous, but just so that they happened in a particular sequence. This more ergonomic interaction will likely affect higher-level forms of experience, and experience over time, as we felt less fatigue from multiple playthroughs when compared to other examples. Perception and memory were still minimally used.

Pump, Pump, and Away tasks players to work together to fill a rocket with air. There is an explicit 10-second time limit to pump air to the rocket before take-off. The players who have made the most inputs (and pumped the most air) win. The context is local team-based cooperative-competitive multiplayer on a standard controller. Unlike previous examples, this game gave us a choice of inputs: either pressing A and B in sequence, or A and Z to the same effect. Having tried Psychic Safari with A and B on the same controller set up, we decided to examine the A and Z experience to see if there was an ergonomic difference. We tried this in three variations: with an NPC teammate against NPCs, with an NPC teammate against humans, and with a human teammate against humans. For the all-NPCs and NPC against human variants, its abilities seemed identical to Psychic Safari. The interesting case was when it was a human team versus a human team. In this case, it performed similarly to Take a Breather. We did find that the A and Z setup felt more natural, as it allowed us to hold the controller in a reasonable way while leaving our fingers resting on both buttons.

Balloon Burst (Figure 11) tasks players to fill a balloon version of Bowser with air. There is an explicit 30-second time limit, however, the challenge can end earlier if players can burst their balloon (i.e., make a sufficient number of inputs). Balloon Burst exists in different contexts depending on the version. The *Mario Party* context is local solo competitive multiplayer on a standard controller. For *Mario Party 2,* it's local team-based cooperative-competitive multiplayer on a standard controller. The mechanisms of interactions (A and B, or A and Z like Pump, Pump, and Away) and goals are identical across these contexts; therefore, the only difference is the individual versus team nature of the two. We played both contexts to compare whether this change affected the competency profile. For *Mario Party 2,* we compared three variations, as we did in Pump, Pump, and Away. We found that it played almost identically to Pump, Pump, and Away; the major difference was that the time limit was not as important. Where previously we would be mashing buttons until the time limit ended, and then waiting for the results, here there was pressure to mash quicker as more efficient inputs meant a shorter game. We similarly played the *Mario Party* context in two variations: against NPCs and against humans. The results were the same. This variable time limit led to more strain on our motor abilities as we tried to push ourselves to beat the other human opponents.

Figure 11: Balloon Burst Gameplay. This figure illustrates a local team-based cooperative-competitive multiplayer alternating input button mashing in the form of Mario Party 2's Balloon Burst.

Ridiculous Relay is a race between a solo player and a three-player group; it has two types of player experiences. Here we focus on the mechanics of the three-player group, particularly the shell section (the first part of the relay). The player must rapidly alternate between the A and B button, which control the right and left oar. The time limit is implicit as the player needs to move fast enough to cover the distance of their segment of the race. When playing, we realized that the three-player team experience was really more like three distinct 1-v-1 experiences put together; as once we were done with our segment of the relay we did not concern ourselves with the performance of others because we could not influence the result. We found that the abilities we used in the shell section seemed the same as previous examples: finger pressing, wrist pointing, attention, perception, and memory.

Speed Skating is a race between the player and three opponents around an Olympic rink. The player skates by alternately pressing the shoulder buttons (L and R). The time limit is implicit and determined by the speed of the player in the lead. The context is local solo competitive multiplayer on a handheld console. Though the controller was different we found the abilities used to be identical to Take a Breather.

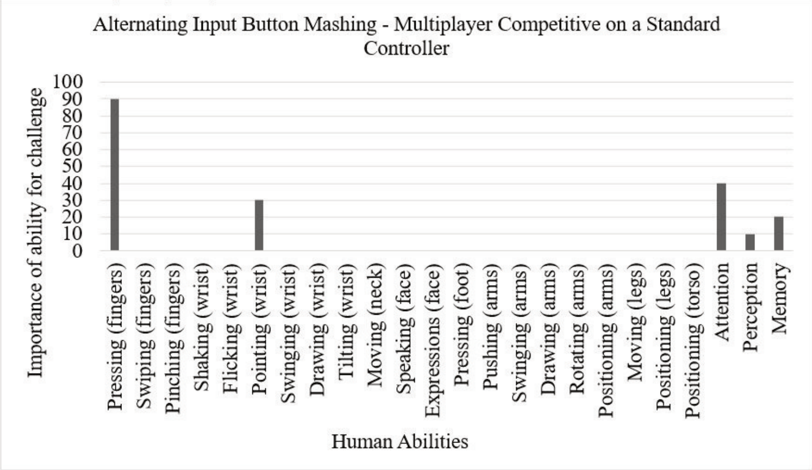
The Colossus of Rhodes fight is the first boss fight in *God of War 2* and is comprised of many smaller challenges (mostly related to combat). During the end of the second phase of the fight, the giant statue grabs the player and, to escape, the player must alternately mash the L1 and R1 buttons. There is an implicit time limit as the player will lose health and potentially die if they cannot escape from the statue. The context is local solo competitive single player on a standard controller. We found the abilities used to be identical to Take a Breather and Speed Skating.

Table 6 shows the competency profile of alternate input button mashing. Although the contexts of the examples differed, we found the abilities used and their amounts, were consistent. Finger pressing continued to be the most important ability (90), thus the limiting ability. Wrist pointing was noticeably used to support finger pressing (30). Attention was used to maintain the sequence, which we felt was slightly more important than wrist pointing (as an incorrect button press could cost us the challenge) so we listed it at 40. Perception was minimally used (10). Memory used varied with the length of the sequence, and so could vary from minimally used to noticeably used, thus we rated it 20.

Table 6

Challenge Description for Alternating Input Button Mashing

Name	Alternating Input Button Mashing
Challenge Information	
Definition	Repeatedly and rapidly press two specific buttons in sequence.
Mechanics	
Button pressing, short time limits, pattern recognition	
Variable components	• Length of sequence/number of buttons pressed • Which buttons are being pressed • Time limit
Context	
Mechanism of Interaction	Pressing buttons
Controller Type	Standard Controller
Number of Players	Multiplayer
Type of Play	Competitive (solo)
Examples	
• Take a Breather, Mario Party 4 • Balloon Burst, Mario Party • Psychic Safari, Mario Party 2	
Variants	
• Controller Type: Handheld console, handheld motion controller (vertical), handheld motion controller (horizontal), keyboard • Number of Players: Single player • Type of Play: Cooperative-Competitive (team based), Cooperative	
Intrinsic Competency Profile	



Source of Difficulty	Finger pressing

CHALLENGE JUTSU: A KNOWLEDGE CAPTURE ARTIFACT

One of our goals is to develop a vocabulary to discuss a player's mechanical experience of a challenge. Challenge descriptions capture the competency profile required for completion, while the player homunculus captures player abilities. We present *challenge jutsu* to link them together.

As a knowledge capture mechanism, challenge jutsu are inspired by the *design patterns* found in HCI (Dearden & Finlay, 2006), and software engineering (Gamma, Helm, Johnson, & Vlissides, 1994). Both are knowledge capture tools with different focuses: design patterns are solution-focused while jutsu are about framing the problem and its root causes. We see design patterns as emerging from challenge jutsu once context of occurrence is fixed.

Challenge jutsu methodology

We document challenge jutsu in three sections: the challenge description, the player description, and the derived mechanical experience.

The challenge description is composed of: a natural language definition of the challenge, the in-game mechanics, the context, examples from mainstream games, context variants, and intrinsic competency profile. These are all needed to narrow down the type of gameplay instance under discussion.

The player description is the player homunculus for the demographic of targeted players. It is represented as a bar graph, with each bar representing proficiency in an associated ability compared to the generic player homunculus. Here we will only discuss generic players, i.e., those the abilities of an able-bodied neurotypical adult, as previously described.

The derived mechanical experience relates the challenge and player descriptions, visualized as a bar graph overlapping the player's abilities with the intrinsic competency profile of the challenge. From this graph, areas of difficulty (where the required ability is greater than the player's ability) are visible and recorded in the jutsu. We identify barriers to challenge completion in order to offer a shortlist of tweaks that could be made to the variable components of the challenge to address the gap between the challenge requirements and the player's abilities.

Challenge jutsu: Single input button mashing

We applied the above methodology to Single Input Button Mashing challenges to detail its challenge jutsu. The challenge description is Table 4, and the player description is the Generic Player Homunculus.

We compared the competency profile with the player description to create the "actual" mechanical experience. When the player's ability is significantly lower, we placed an X above the column to indicate that it is highly unlikely the player will be able to successfully interact complete the given challenge. We call these *unintentional sources of difficulty*. We then attempted to provide ways to adjust the challenge to accommodate for the player's abilities.

When the player's abilities are marginally lower or higher than the competency profile, we used an exclamation point (!). Exclamation point (!) abilities may not affect the player's chances of success. Players may compensate for these abilities by using supporting abilities, or alternatively can train these abilities for these contexts. We called these identified areas *potential sources of difficulty* as their effect on play varies from player to player.

For player abilities that greatly exceed the competency profile we place a check mark to indicate that they can easily complete that component of the challenge. If all player abilities are well above the required competency profile, this challenge is probably too easy and may bore the player. But this is something for future work.

After all of this we composed it into a final challenge jutsu of a single input button mashing on a standard controller for generic player homunculus, represented in Tables 7 and 8.

Variants in challenge jutsu

Consider a single input button mashing challenge played on a standard controller versus an arcade machine. Both scenarios have physical buttons and are identical in the game context, except for the mechanism of interaction. An arcade machine has larger buttons, which affords pressing with the whole hand and arm rather than just fingers. Rather than creating new jutsu when the only change is the controller, we instead use *jutsu variants*. *Jutsu variants* are when the same challenge exists across different motor abilities, for example, substituting finger pressing with foot pressing or arm pressing. Variants do not change the shape of the competency profile (how much each ability is used), it just replaces the dominant motor ability. Consider the case of the 100m Dash in *Mario and Sonic at the Rio 2016 Olympic Games: Arcade Edition* (Sega and Racjin, 2016) which is an alternating input button mashing challenge presented as a gross motor challenge, replacing finger pressing and wrist pointing with foot pressing and leg moving (Tables 9 and 10).

Table 7

Challenge Jutsu for Single Input Button Mashing – Challenge Description

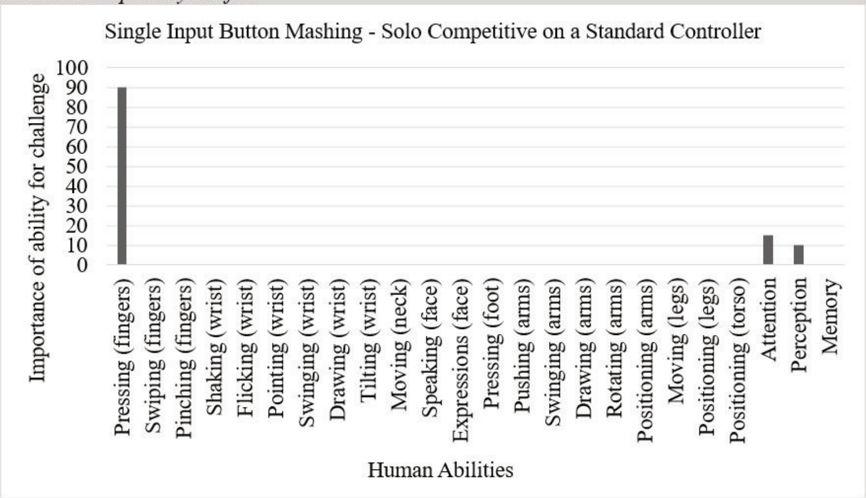
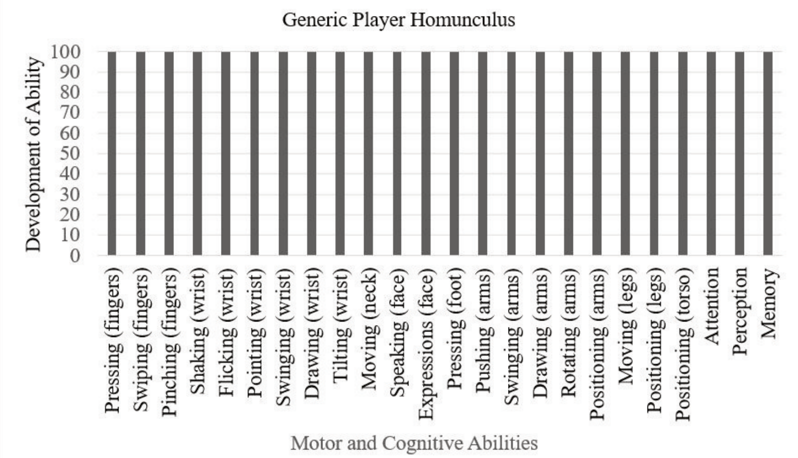
Name	Single Input Button Mashing
Challenge Information	
Definition	Repeatedly pressing a specific single button or key on the controller as fast as possible within a given time limit.
Mechanics	
Button pressing, short time limits	
Variable components	• Time limit
Context	
Mechanism of Interaction	Pressing buttons
Controller Type	Standard Controller
Number of Players	Single player
Type of Play	Competitive (solo)
Examples	
• Dragon's Breath, South Park: The Stick of Truth • Torture Attacks, Bayonetta and Bayonetta 2 • Boss Knockouts, Donkey Kong Country: Tropical Freeze	
Variants	
• Controller Type: Handheld console, handheld motion controller (horizontal), keyboard • Number of Players: Multiplayer • Type of Play: Cooperative-Competitive (team based), Cooperative	
Intrinsic Competency Profile	
	
Source of Difficulty	Finger pressing

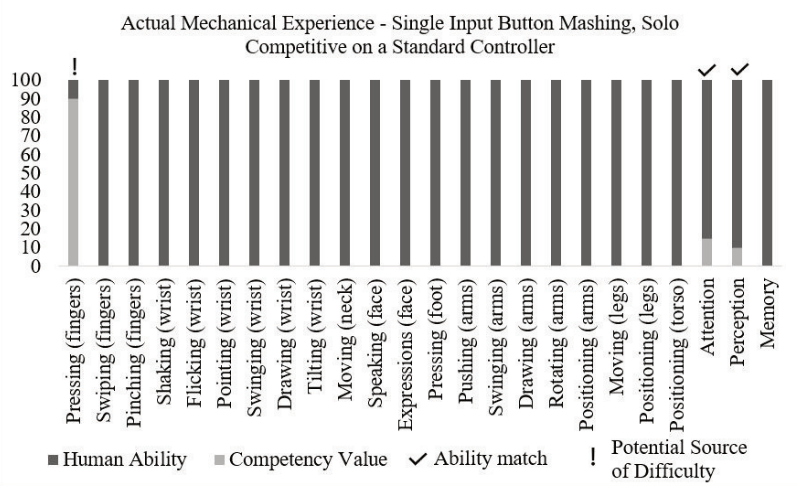
Table 8

Challenge Jutsu for Single Input Button Mashing – Player Model and Mechanical Experience

Name	Single Input Button Mashing
Player Profile	



Mechanical Experience	



Areas of difficulty	• N/A
Suggested Tweaks	• N/A

Table 9

Challenge Jutsu for Alternating Input Button Mashing – Challenge Description

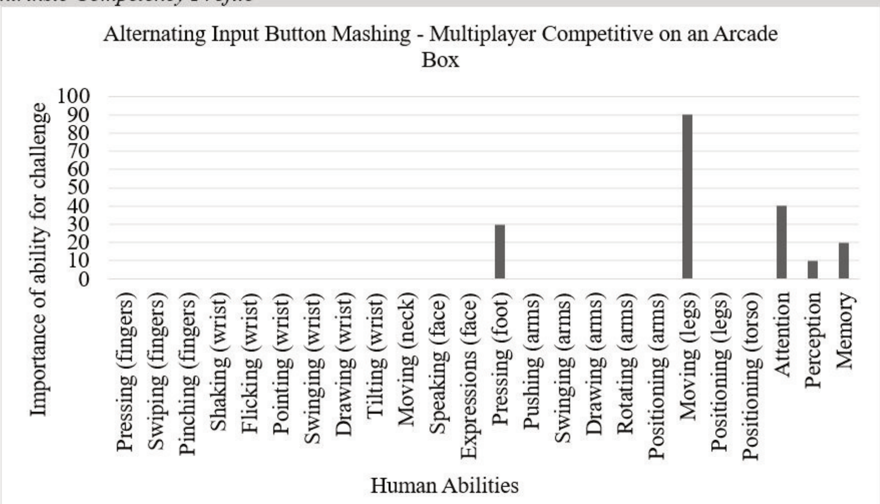
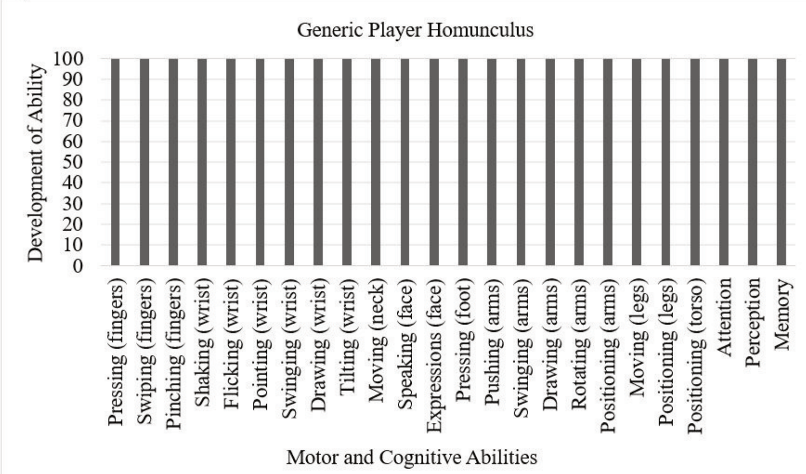
Name	Alternating Input Button Mashing
Challenge Information	
Definition	Repeatedly pressing a specific single button or key on the controller as fast as possible within a given time limit.
Mechanics	
Button pressing, short time limits	
Variable components	• Time limit
Context	
Mechanism of Interaction	Pressing buttons
Controller Type	Arcade Box
Number of Players	Single player
Type of Play	Competitive (solo)
Examples	
•	
Variants	
• Controller Type: Handheld console, handheld motion controller (horizontal), keyboard • Number of Players: Multiplayer • Type of Play: Cooperative-Competitive (team based), Cooperative	
Intrinsic Competency Profile	
	
Source of Difficulty	Arm pressing

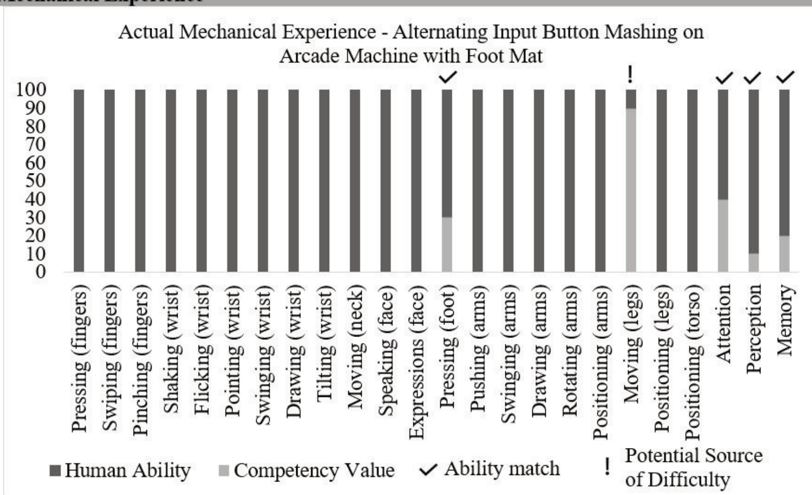
Table 10

Challenge Jutsu for Alternating Input Button Mashing – Player Model and Mechanical Experience

Name	Alternating Input Button Mashing
Player Profile	



Mechanical Experience	



Areas of difficulty	• N/A
Suggested Tweaks	• N/A

Collection and organization of challenge jutsu

An organized system of jutsu should be helpful to understand individual challenges, but their sheer number is a bit overwhelming. Furthermore, as collecting these is ongoing, and will hopefully become a collaborative community effort, we need a proper system to organize this information. We propose an online, public database. This way, via different views, we can accommodate both user groups: researchers looking to analyze existing games, and designers looking to create new games. A structured wiki may provide the easiest access to both user groups.

Organizing the jutsu in a manner useful for different user groups is the biggest challenge. Our two primary user groups (researchers, and designers) have different use cases, and thus need different views. We dub these the *analysis view* and the *design view*. We expect to add more views as needs arise in the future.

A primary analysis view might sort the jutsu by their source of difficulty as Figure 12 shows. When other dimensions of game analysis that fit the jutsu pattern become clear, other views can be added. The analysis view is useful for understanding interaction barriers (and thus accessibility problems). For example, investigating whether a game is playable by children with cerebral palsy, a researcher would know the specific abilities of their participants, but may not know enough about gaming to identify what games would be playable without playing it themselves. The analysis view sorts the challenges by abilities, highlighting of what challenges are achievable by their chosen players.

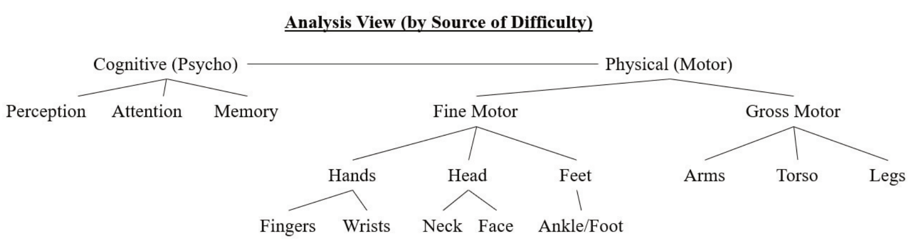


Figure 12: Analysis-View organization of Challenge Jutsu Database. This figure illustrates an organization of identified challenge jutsu based on their source of difficulty.

A design view might sort the jutsu by their challenge types as Figure 13 shows. When crafting new games, designers tend to discuss in terms of game concepts like (types of) challenges rather than abilities used. This view meshes designer's mental model of game creating, but additionally reveals ability information.

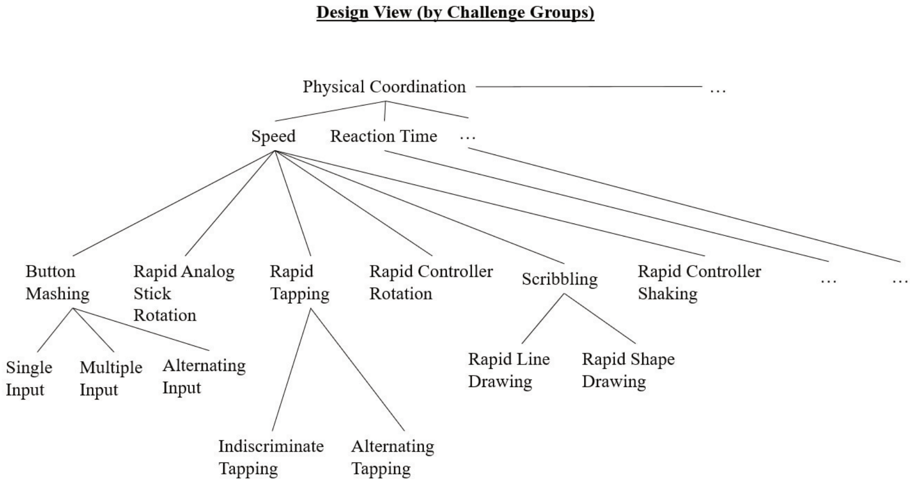


Figure 13: Design-View organization of Challenge Jutsu Database. This figure illustrates an organization of identified challenge jutsu based on the challenge types.

DISCUSSION AND CONCLUSION

We created the *Jutsu Framework (JF)* to analyze and discussed the mechanical player experience.

Specifically, we:

- Described a generic player model based on cognitive and motor abilities;
- Provided a methodology to define gameplay challenges via their competency profiles;
- Outlined a standardized structure for presenting *challenge jutsu;*
- Presented a methodology for how to create *challenge jutsu*; and
- Presented multiple methods of organization for challenge jutsu.

Our presentation of challenge jutsu does not aim to be comprehensive.

Rather, our main contribution is the rationale and methodology behind their construction and organization, coupled with sufficiently detailed examples to evaluate its effectiveness.

Potential applications

The first use for JF is in helping to identify and quantify the sources of mechanical difficulty. This enables creative ways to let designers compensate for player differences between their abilities and the challenge competency profiles. This could also support existing research into accessible controller design or adaptive gaming. This tool can also help support a more targeted form of user testing. Designers know their intended market; challenge competency profiles and specific demographic abilities can drive the selection of test cases, hopefully leading to reduced testing time around playability.

Challenge jutsu also can foster systematic exploration of why challenges work in specific contexts and not others. For example, by understanding the cognitive and motor abilities used in challenges, it may become obvious that certain abilities do not translate well to contexts like VR. Or, alternatively, why certain challenges work for some of the target demographic and not others. This framework can support game balancing and adaptive gaming research into dynamic difficulty adjustment. It could give dynamic difficulty adjustment frameworks a way to consider that the problem is with the game design, not the individual player's abilities.

Relatedly, JF provides a pathway for exploring novel challenges through jutsu variants. As gaming continues to grow as an industry, novelty becomes more difficult to achieve and more important as a selling feature. The structured presentation of the jutsu makes underrepresented and unused abilities salient. This gives designers a guide to explore that space and create new challenges.

Ambitiously, we hope that a further application will be a more concrete rating system for games based on their playability and not their aesthetic context. If game ratings can be upfront about their accessibility requirements, this would make gaming a more inviting and available hobby to people with disabilities. It would allow for gamers to connect with games that are mechanically appropriate for them and incentivize designers to think about creating more mechanically inclusive games.

Future work

We plan to work on further refinements, and then experimental validation. Expanding the work to encompass Adams' (2010) full list, as refined through our methodology, will create many more challenge jutsu. We also need to define more player homunculi for various demographics. The player homunculi also need refinement to expand the cognitive aspect. Validation will require running a series of experiments to first confirm our challenge competency profiles from our close readings. We could then test the relationship between competency profiles and various player homunculi to get a better understanding of the playability of different challenges in different contexts.

While our focus here is on jutsu based on mechanical experience, the concept can extend to all aspects. Each is based on different player characteristics; for example, the socio-cultural experience requires an analysis of the players' knowledge of society and culture. Challenge descriptions would require adjustments to incorporate related elements, but we do not foresee the fundamental structure changing.

Understanding the mechanical experience of game challenges is a necessary first step in dealing with accessibility issues. This work is ready to expand and we plan to put all our data online shortly for just that purpose. We see our foundational work as a starting point to systematically address issues in design accessibility to improve the player experience for many under-served gamer demographics.

REFERENCES

Adams, E. (2010). *Fundamentals of game design* (2nd ed.). Berkeley, CA: New Riders.

Bartle, R. (1996). Hearts, clubs, diamonds, spades: Players who suit MUDs. *Journal of MUD Research,* 1-19.

Bateman, C., & Boon, R. (2005). *21st century game design (Game Development Series).* Charles River Media, Inc.

Bateman, C., Lowenhaupt, R., & Nacke, L. (2011). Player typology in theory and practice. *Proceedings of DiGRA* 2011. Hilversum.

Bjork, S., & Holopainen, J. (2004). *Patterns in game design*. Needham Heights, MA: Charles River Media.

Bungie. (2001, November 15). *Halo: Combat Evolved*. Xbox. Washington: Microsoft Game Studios.

Capcom. (2017, January 24). *Resident Evil 7: Biohazard*. Playstation 4. Osaka, Japan: Capcom.

Card, S. K., Moran, T. P., & Newell, A. (1983). *The psychology of human-computer interaction*. Hillsdale, New Jersey: L. Erlbaum Associates Inc.

Cawthon, S. (2014, August 8). *Five Night's At Freddy's*. Windows. Houston, Texas: Steam.

Climax Studios. (2009, December 8). *Silent Hill: Shattered Memories*. Wii. Tokyo, Japan: Konami.

Cooking Mama Ltd. (2006, September 12). *Cooking Mama*. DS. New Jersey: Majesco Entertainment.

Cooking Mama Ltd. (2007, March 20). *Cooking Mama*: *Cook Off*. Wii. New Jersey: Majesco Entertainment.

Dearden, A., & Finlay, J. (2006). Pattern languages in HCI: A critical review. *Human-Computer Interaction, 21*(1), 49-102. doi:10.1207/s15327051hci2101_3

Djaouti, D., Alvarex, J., Jessel, J., Methel, G., & Molinier, P. (2008). A gaemplay definition through video game classification. *International Journal of Computer Games Technology, 4*.

Drachen, A., Canossa, A., & Yannakakis, G. (2009). Player modeling using self-organization in Tomb Raider: Underworld. *Proceedings of the IEEE Symposium on Computational Intelligence and Games*. Milan.

Feil, J., & Scattergood, M. (2005). *Beginning game level design*. Boston, MA: Thomson Course Technology.

Fleishman, E., Quaintance, M., & Broedling, L. (1984). *Taxonomies of human performance: The description of human tasks*. San Diego, CA: Academic Press.

Frictional Games. (2010, September 8). *Amensia: The Dark Descent*. Windows. Helsingborg, Sweden: Frictional Games.

Gaijin Games. (2013, Februrary 26). *Bit.Trip Presents...Runner 2: Future Legend of Rhythm Alien*. WiiU. California: Aksys Games.

Gamma, E., Helm, R., Johnson, R., & Vlissides, J. (1994). *Design patterns: Elements of reusable object-oriented software.* U.S.: Addison-Wesley.

Godde, B., & Voelcker-Rehage, C. (2017). Cognitive resources necessary for motor control in older adults are reduced by walking and coordination training. *Frontiers in Human Neuroscience, 11,* 156.

Hamari, J., & Tuunanen, J. (2014). Player types: a meta-synthesis. *Transactions of the Digital Games Research Association, 1,* pp. 29-53. doi:10.26503/todigra.v1i2.13

Harmonix. (2007, November 20). *Rock Band*. Xbox 360. New York City, New York, United States of America: MTV Games.

Hu Wen Zeng, and Cheetah Games. (2019, August 23). *Piano Tiles 2 (Don't Tap...2).* Android. Hong Kong: Google Play Store.

Hudson Soft. (1999, February 8). *Mario Party*. Nintendo 64. Kyoto, Kyoto, Japan: Nintendo.

Hudson Soft. (2000, January 24). *Mario Party 2*. Nintendo 64. Kyoto, Japan: Nintendo.

Hudson Soft. (2001, May 7). *Mario Party 3*. Nintendo 64. Kyoto, Japan: Nintendo.

Hudson Soft. (2002, November 8). *Mario Party 4.* Gamecube. Kyoto, Japan: Nintendo.

Hudson Soft. (2002, November 8). *Mario Party 5*. Gamecube. Kyoto, Kyoto, Japan: Nintendo.

Hudson Soft. (2007, May 29). *Mario Party 8.* Wii. Kyoto, Japan: Nintendo.

Hunicke, R., LeBlanc, M., & Zubek, R. (2004). MDA: A formal approach to game design and research. *Proceedings of the AAAI Workshop on Challenges in Game AI, 4,* 17-22.

id Software. (1996, June 22). *Quake.* Windows. New York, New York: GT Interactive.

Konami. (1983). *Track & Field. Arcade*. Tokyo, Japan: Konami.

Konami. (1999, March). *Dance Dance Revolution*. Arcade. Tokyo, Japan: Konami.

Level-5. (2009, August 24). *Professor Layton and the Diabolical Box*. Kyoto, Kansai, Japan: Nintendo.

McMahon, N., Wyeth, P., & Johnson, D. (2015). From challenges to activities: Categories of play in videogames. *Proceedings of the 2015 annual symposium on computer-human interaction* (pp. 637-642). New York: ACM. doi:10.1145/2793107.2810333

Namco. (2004, September 27). *Donkey Konga*. Gamecube. Kyoto, Kyoto, Japan: Nintendo.

NDcube. (2018, October 5). *Super Mario Party*. Switch. Kyoto, Japan: Nintendo.

Nintendo EAD. (2011, November 29). *Legend of Zelda: Skyward Sword*. Wii. Kyoto, Kyoto, Japan: Nintendo.

Nintendo EAD. (2014, May 30). *Mario Kart 8*. WiiU. Kyoto, Kyoto, Japan: Nintendo.

Nintendo EAD. (2017, October 27). *Super Mario Odyssey*. Switch. Kyoto, Kyoto, Japan: Nintendo.

Nintendo EAD Group No. 2. (2008, November 17). *Animal Crossing*: *City Folk*. Wii. Kyoto, Kyoto, Japan: Nintendo.

Nintendo EAD Group No. 2. (2009, July 26). *Wii Sports Resort*. Wii. Kyoto, Kyoto, Japan: Nintendo.

Nintendo SPD Group No. 4, & ND Cube. (2010, October 3). *Wii Party*. Wii. Kyoto, Japan: Nintendo.

Obsidian Entertainment, & South Park Digital Studios. (2014, March 4*). South Park: The Stick of Truth.* Xbox 360. Rennes, France: Ubisoft.

Pajitnov, A., & AcademySoft. (1986). *Tetris*. MS-DOS. North America: AcademySoft.

Papegaaij, S., Taube, W., Baudry, S., Otten, E., & Hortobagyi, T. (2014). Aging causes a reorganization of cortical and spinal control of posture. *Frontiers in Aging Neuroscience, 6,* 28.

Penfield, W., & Rasmussen, T. (1950). *The cerebral cortex of man*. New York: Macmillan.

Perfect Tap Games. (2017, April 28). *Chicken Scream*. Android. Dubai, United Arab Emirates: Perfect Tap Games.

Platinum Games. (2010, January 5). *Bayonetta*. Xbox 360. Tokyo, Japan: Sega.

Platinum Games. (2014, October 24). *Bayonetta 2*. WiiU. Kyoto, Kyoto, Japan: Nintendo.

Retro Studios. (2010, November 21). *Donkey Kong Country Returns*. Wii. Kyoto, Kyoto, Japan: Nintendo.

Retro Studios. (2014, February 21). *Donkey Kong Country*: *Tropical Freeze*. WiiU. Kyoto, Kyoto, Japan: Nintendo.

Schell, J. (2014). *The art of game design: A book of lenses* (2nd ed.). Boca Raton, Florida: CRC Press.

Sega and Racjin. (2016, June). *Mario and Sonic at the Rio 2016 Olympic Games Arcade Edition*. Arcade. Kyoto, Japan: Sega.

Sega Sports R&D. (2007, November 6). *Mario and Sonic at the Olympic Games*. Wii. Tokyo, Japan: Sega.

Sega Sports R&D. (2008, January 22). *Mario and Sonic at the Olympic Games*. DS. Tokyo, Japan: Sega.

Sega Sports R&D and Racjin. (2009, October 13). *Mario and Sonic at the Winter Olympic Games*. Wii. Tokyo, Japan: Sega.

Sega Sports R&D, & Arzest. (2016, March 18). Ma*rio and Sonic at the Rio 2016 Olympic Games*. 3DS. Kyoto, Japan: Nintendo.

Sega Sports R&D, & Arzest. (2016, June 24). *Mario and Sonic at the Rio 2016 Olympic Games*. WiiU. Kyoto, Japan: Nintendo.

Sega Sports R&D, & Racjin. (2009, October 13). M*ario and Sonic at the Winter Olympic Games*. DS. Tokyo, Japan: Sega.

Sega Sports R&D, & Racjin. (2011, November 15). *Mario and Sonic at the London 2012 Olympic Games*. Wii. Tokyo, Japan: Sega.

Sega Sports R&D, & Racjin. (2012, February 14). *Mario and Sonic at the London 2012 Olympic Games*. 3DS. Tokyo, Japan: Sega.

Seidler, R. D., Bernard, J. A., Burutolu, T. B., Fling, B. W., Gordon, M. T., Gwin, J. T., . . . Lipps, D. B. (2010). Motor control and aging: Links to age-related brain structural, functional, and biochemical effects. *Neuroscience & Behavioural Reviews, 34*(5), 721-733.

Sleeping Beast Games. (2012, December 1). *Spaceteam*. Android. Montreal, Quebec, Canada: Sleeping Beast Games.

Sony Computer Entertainment Santa Monica Studio. (2007, March 13). *God of War 2.* Playstation 2. California, United States of America: Sony Computer Entertainment.

Square Enix, and Monolith Soft. (2006, August 15). *Final Fantasy VII: Dirge of Cerberus*. Playstation 2. Tokyo, Japan: Square Enix.

Stewart, B. (2011). Personality and play styles: A unified model. Retrieved from *Gamasutra*: http://www.gamasutra.com/view/feature/6474/personality_and_play_styles_a_.php

Tseng, F.-C. (2010). Segmenting online gamers by motivation. *Expert Systems with Applications, 38,* 7693-7697.

Ubisoft Montreal. (2010, October 24). *Shaun White Skateboarding*. Wii. Rennes, France: Ubisoft.

Ubisoft Paris and Ubisoft Milan. (2009, November 1). *Just Dance*. Wii. Rennes, France: Ubisoft.

UltraRu. (2015, July 9). *Impossible Jump*. Android. Google Play Store.

Veli-Matti, K. (2014). Puzzle is not a game! Basic structures of challenge. *Proceedings of the 2013 DiGRA International Conference Defragging Game Studies*.

Yee, N. (2007). Motivations of play in online games. *Journal of CyberPsychology and Behaviour, 9,* 772-775.

Zackariasson, P., Wahlin, N., & Wilson, T. (2010). Virtual identities and market segmentation in marketing in and through Massively Multiplayer Online Games (MMOGs). *Services Marketing Quarterly, 31,* 275-295.

Zhu, T., Wang, B., & Zyda, M. (2018). Exploring the similarity between game events for game level analysis and generation. *Proceedings of the 13th International Conference on the Foundations of Digital Games, 8*, 1-7. New York: ACM. doi:10.1145/3235765.3235804