Towards Accessibility of Games: Mechanical Experience, Competence Profiles and Jutsus.

Abstract

Accessibility of games is a multi-faceted problem, one of which is providing mechanically achievable gameplay to players. Previous work focused on adapting games to the individual through either dynamic difficulty adjustment or providing difficulty modes; thus focusing on their failure to meet a designed task. Instead, we look at it as a design issue; designers need to analyse the challenges they craft to understand why gameplay may be inaccessible to certain audiences. The issue is difficult to even discuss properly, whether by designers, academics or critics, as there is currently no comprehensive framework for that. That is our first contribution. We also propose challenge jutsus – structured representations of challenge descriptions (via competency profiles) and player models. This is a first step towards accessibility issues by better understanding the mechanical profile of various game challenges and what is the source of difficulty for different demographics of players.

1.0 Introduction

Different Types of Experience

When discussing, critiquing, and designing games, we are often concerned with the "player experience" – but what this means is unsettled as games are meant to be consumed and enjoyed in various ways. Players can experience games *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). Each aspect corresponds to different ways that the player engages with the game. We can map the different forms of experience to the Eight Types of Fun (Hunicke, LeBlanc, & Zubek, 2004) (Table 1). The "player experience" is the combination of these different modes of experience.

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

Table 1: Hunicke, LeBlanc, and Zubek's 8 Types of Fun.

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 (mechanics, aesthetics, story, technology), the Experiential Tetrad shows the aspects of experience. These aspects are defined by how the individual's abilities and knowledge relate with the game elements. We organise the Experiential Tetrad to show how different viewpoints change the entry-point to experience. *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. The *mechanical experience* is a relation between a player's physical and cognitive abilities, and gameplay challenges.

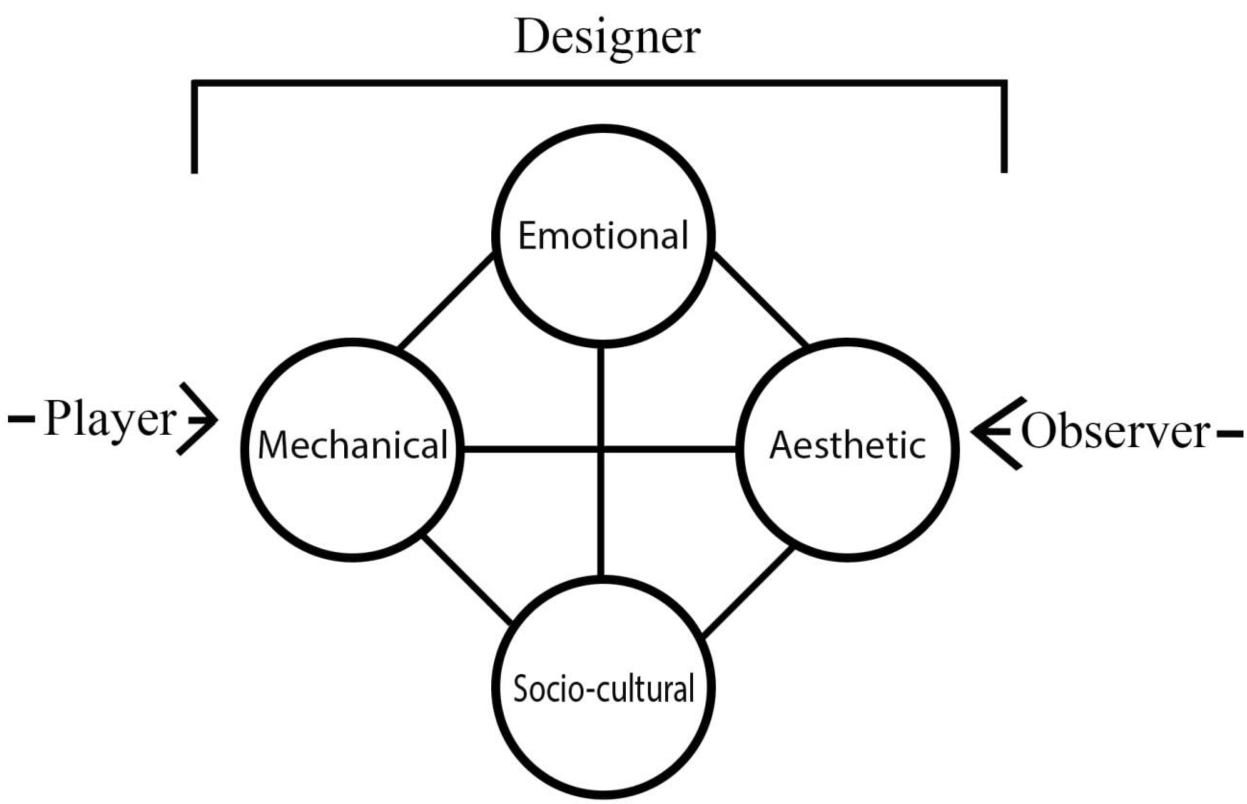


Image 1: Player Experience as a Tetrad with Player, Observer, and Designer viewpoints.

Crafting a game relies on understanding and managing how the different aspects relate to each other. Consider horror games: the player experience is primarily emotional, supported by aesthetic and socio-cultural experiences. Mechanics should 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 Amensia: The Dark Descent (Frictional Games, 2010) are good examples: the control mechanics are simple navigation and object interaction. The simplicity of the mechanics 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. Witness expert players of games like Five Nights at Freddy's

(Cawthon, 2014) or Resident Evil 7: Biohazard (Capcom, 2017) where the emotional experience has become secondary to just beating the game.

Gameplay is a two-way interaction between a player and a game system; a successful interaction happens when the player's ability levels match up with what the game is asking them to do. To design more reliably successful interactions, it is useful to understand the two components of interaction: the player's abilities and the tasks the game sets for them to complete.

If a player cannot provide adequate inputs to the game, they will not be able to make progress. This makes it rather difficult for them to engage with the emotional, aesthetic, or socio-cultural aspects of the game. Similarly, a vision-impaired player may not be able to appreciate the subtleties of the aesthetic experience, and so on for the other aspects. Thus, understanding the competency profile of each experiential aspect can help designers adapt their game to different audiences that may display different abilities, knowledge, cultural background, etc.

Here we focus on the *mechanical experience* and explore how a game's challenges can act as a barrier to different groups of players. We chose to start there, as it seems to the simplest to study, especially via the scientific method. Understanding these design problems and barriers should enable us to create tools to address them.

Mechanical Experience

Specifically, we understand the *mechanical experience* as the player's motor and cognitive interactions with the gameplay. It is crafted by the designer through the selection of game mechanics and challenges. Challenges and player abilities combine to create the mechanical player experience, based on whether the challenge is achievable or not. The goal is to create *mechanically achievable gameplay* – where the mechanical experience is interesting in and of itself to the target audience, or invisible when it is meant to be a supporting actor in the overall experience. A player who cannot progress in the game because the mechanics are beyond their abilities will not be able to experience the game aesthetically, emotionally, or socio-culturally.

Designing challenges that test the player's motor and cognitive abilities, while remaining achievable, is complex. Currently design practice assumes that players have normative abilities and are unencumbered during play. This implicitly omits certain groups of players from the challenges they design. Consider Just Dance 2019 (Ubisoft Paris, Ubisoft Pune, Ubisoft Shanghai, 2018), a dancing and rhythm game where players mirror the movements of on-screen dancers. Though the game is rated "E for Everyone", gamers with motor impairments cannot engage with the game in the same way as able-bodied players. Many songs ask the players to rapidly move their arms back and forth; possible yet challenging for able bodied games, an elderly player with limited mobility in their joints, may find it impossible. Did the designers and the marketers consider this when labelling it for "Everyone"?

This is a question of accessibility. Who are games designed for, and how do we deliver mechanically meaningful experiences to players of differing abilities? The release of Sekiro: Shadows Die Twice (FromSoftware, 2019) brought these issues to the forefront. The game is known as being unforgiving towards its players, rewarding those who can master its mechanics with a feeling of accomplishment – amongst able-bodied players. The discussion centred around the idea of mechanical difficulty (how challenging it is to "beat" parts – or all – of the game), and whether adding accessibility options was counter to the intended experience.

The question then becomes: by certain mechanics and challenges, does the (mechanical) experience of the player line up with the intended experience? Are games designed with, and tested by, a sufficiently broad set of gamers? Do designers even have the knowledge and tools to achieve this? The barriers in games like Just Dance and Sekiro are not necessarily intentional – they may represent unintentional difficulty in the challenges. Designers want their experiences to be challenging but possible.

What Can We Do?

We need to understand the relationship between the mechanical experience *as designed* and as *experienced by* the players. There is currently no framework for designers, players, and critics to talk about this – and this is where we begin our work.

Mechanical experience design can be viewed as a task modeling problem. Games are composed of various gameplay challenges that the player must successfully complete to win; this is inline with many other researchers' view of the relationship between challenges and games (Adams, 2010; Djaouti, Alvarex, Jessel, Methel, & Molinier, 2008; Feil & Scattergood, 2005; McMahon, Wyeth, & Johnson, 2015; Veli-Matti, 2014). If we can model gameplay challenges as tasks, we can then describe them in terms of the abilities that players need to successfully complete them. This is in line with Fleishman, Quaintance, and Broedling's idea of a *competency profile* – the set of cognitive and motor abilities that characterize a task (Fleishman, Quaintance, & Broedling, 1984). This would give us a vocabulary for mechanical difficulty, and from there be able to discuss ensuing accessibility issues.

Necessary abilities would let us to model the *expected mechanical experience* – what the designers are trying to create. Then *actual mechanical experience* can be modelled by how the player's personal abilities compare to the ones required to complete the tasks. By understanding both, we get a way to design for accessibility and discuss areas where changes are needed.

To help compare the expected and actual mechanical experience, we assemble all the required information into a structure that we call a *challenge jutsu*. This structure can be used 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. The rest of this paper will delve more deeply into competency profiles and player abilities before exploring the construction of challenge jutsus.

Our Contribution

We create *challenge jutsus* – a method to organize the knowledge of challenge competency profiles along with player profiles and the sensitivity of each to small change in game mechanics. This provides a tool to predict how the *mechanical experience* will change for different player profiles. The effectivity of challenge jutsus crucially relies on understanding the *mechanical difficulty source* of each challenge.

Specifically, we:

- Provide a methodology to define gameplay challenges via their competency profiles,
- Describe a generic player model based on cognitive and motor abilities,
- Outline a standardized structure for presenting *challenge jutsus*,
- Present a methodology for how to create *challenge jutsus*, and
- Present multiple methods of organization for challenge jutsus.

While our focus here is on jutsus based on mechanical experience, the concept can be applied 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.

Our presentation of challenge jutsus 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.

3.0 The Player

Players – people – are incredibly complex to model. Player typologies are common in game studies to understand motivations for play and player satisfaction. The main idea is to identify player archetypes based on sets of psychometric (Bateman & Boon, 2005; Tseng, 2010; Stewart, 2011; Zackariasson, Wahlin, & Wilson, 2010; Bateman, Lowenhaupt, & Nacke, 2011) or in-game behavioural characteristics (Bartle, 1996; Yee, 2007; Drachen, Canossa, & Yannakakis, 2009). Analyses and criticisms regarding the usefulness and validity of typologies are numerous (Bateman, Lowenhaupt, & Nacke, 2011; Hamari & Tuunanen, 2014). Suffice it to say they do not suit our purposes; they are focused on understanding why players make decisions in games, not how they may physically interact with a game. We need to model the psycho-motor aspects, in other words the *mechanical player*.

Information processing-based user models in HCI align closely with our goals. They divide a user into subsystems in order to evaluate processing bottlenecks and limitations. The Model Human Processor (Card, Moran, & Newell, 1983) (MHP) divides a generic user into three subsystems: motor, cognitive, and memory. They model each subsystem as having its own processor to handle tasks unique to that system and assume that all systems are used to complete a given task, to see which systems are the bottleneck for task completion. We adopt this approach. However, the MHP's subsystems are too abstract for our purpose, and we will need to refine them.

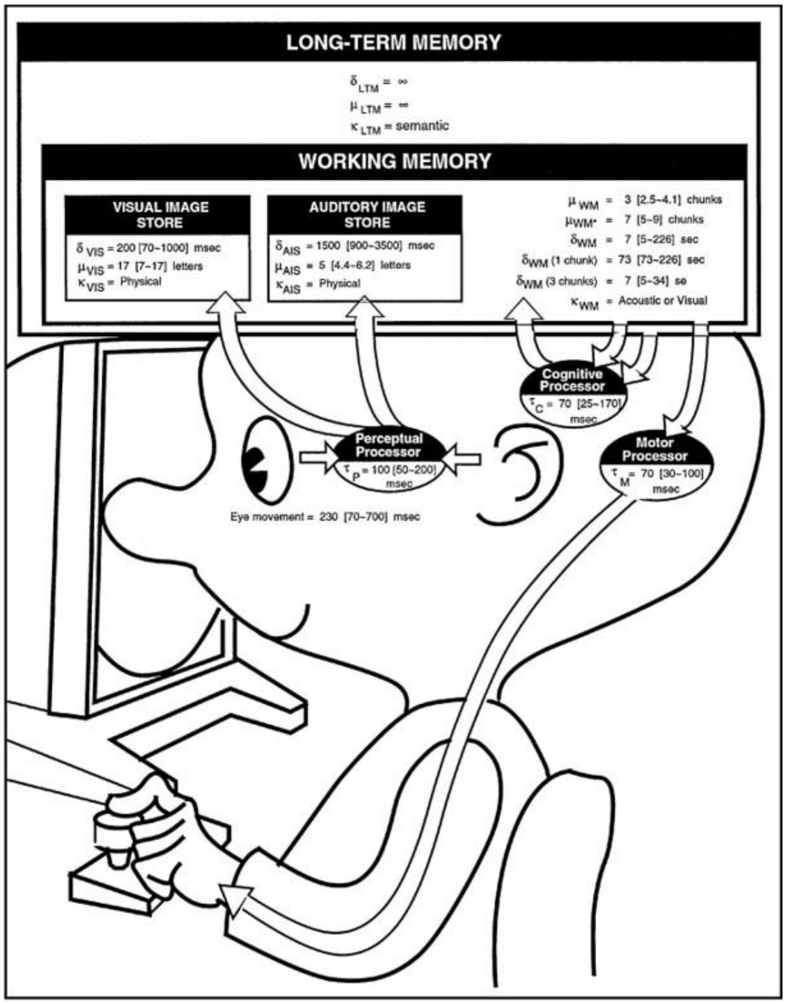


Image 3.1: The Model Human Processor (Card, Moran Newell).

To the best of our knowledge, the MHP viewpoint has not been applied to players. Here we will detail the motor system as used to interact with games. We can then analyse 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 normative player, we will use the term *Player Homunculus* – an abstract representation of a player with normative motor and cognitive abilities. 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.

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, as many seemingly motor-based challenges end up bottlenecked by lower-level cognitive processing abilities. For the current homunculus iteration, we limit these to *perception*, *attention*, and *memory*, in line with the MHP model of users.

We first look at controllers as all game interactions are mediated by them. We focus on standard controllers (e.g. Xbox 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 generate a list of possible interactions (e.g. press button, pull trigger, shake controller) for each controller. We abstract from these interactions to the movement that drives it, so "pressing a button" becomes "pressing" (Table 3.1). We further classify these abilities into fine or gross motor abilities, but indicate when an action could reasonably fall into both categories. This list is not comprehensive, as new controllers and modes of interaction will appear; however, this current set covers a wide variety of games and control types and so should serve as a reasonable starting point.

Motor Actions	Hardware Context
Pressing	Standard controllers, handheld motion controllers, handheld consoles, keyboards, fight sticks, mat controllers
Bumping	Standard controllers, handheld consoles
Pulling	Standard controllers, handheld motion controllers
Moving	Standard controller, handheld motion controller, full body motion controller, handheld console, fight sticks, mice
Swiping	Standard controller, smartphones/tablets
Pinch-to-zoom	Standard controllers, smartphones/tablets
Swinging	Handheld motion controller
Pointing	Handheld motion controller
Shaking	Handheld motion controller, smartphones/tablets, handheld consoles
Drawing	Handheld motion controller, smartphones/tablets, handheld consoles
Thrusting	Handheld motion controller
Tilting	Handheld motion controller, smartphones/tablets, handheld consoles
Flicking	Handheld motion controller, smartphones/tablets
Positioning	Full body motion controller
Tapping	Smartphones/tablets, handheld consoles
Speaking	Smartphones/tablets, handheld consoles
Making facial expressions	Handheld consoles
Clicking	Mice
Scrolling	Mice

Table 3.1: Motor interactions available for each controller type. Colour codes: fine (blue) , gross (yellow), both (green).

Refining the Homunculus for Games

Redundant actions need to be eliminated, and abstract interactions refined to concrete motor abilities. We thus separate actions by the body parts used in the motion and the context in which it is used. This lets us combine actions that are similar into specific groups. Combining the body part (e.g. finger) and action (e.g. pressing) into (e.g. finger pressing) create motor abilities.

As an example of our reasoning, we show our refinement of fine motor abilities. As the full list is long, the details for the gross motor abilities have been omitted for scope. However, we will be including the final list of gross motor abilities alongside the fine motor to illustrate the eventual size of this project.

Fingers

One approximation is that pressing, bumping, pulling, tapping, and clicking 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, 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 versus 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, and 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 their 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. We use Newell's Time Scale of Human Action to determine reasonably similar times. Newell outlines different "bands" (social, rational, cognitive, biological) which describes the order of magnitudes in which different actions happen (Mackenzie, 2013). The actions here all fall in the cognitive band (on the scale of hundreds of milliseconds to tens of seconds), and so can reasonably be considered equivalent from a processing standpoint. Therefore, we consider flicking and swiping to be the same (though an even finer model could separate them). Scrolling is where the player bends their 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. Pressing is an entirely adductive movement (your finger is always moving inwards/towards your body), while scrolling is both abductive and adductive (you can move the scroll wheel backward - towards your body - or forward - away from your body). Swiping is both abductive and adductive, and thus a closer fit 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 focus on motor coordination, which can be measurably more difficult for different age groups (performing coordinated activities has been shown to increase cognitive load for older adults (Malcolm, Foxe, Butler, & De Sanctis, 2015; Seidler, et al., 2010; Papegaaij, Taube, Baudry, Otten, & Hortobagyi, 2014; Lindenberger, Marsiske, & Baltes, 2000; Godde & Voelcker-Rehage, 2017).

We distinguish between single task coordinated actions (STCA), like pinch-to-zoom, and multi-task coordinated actions (MTCA). STCAs require movement coordination to accomplish a single specified goal, for example pressing tiles in Piano Tiles 2 (Hu Wen Zeng, and Cheetah Games, 2019) using your thumbs 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, but with different 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. Nevertheless, there seems to be enough in-game difference to keep some wrist movements as separate actions.

Pointing is the controlled movement of the wrist (mainly), used in the context of positioning a cursor using a handheld motion controller. Here wrist movements are limited to lateral (wrist flexion and extension Image 3.1 – like waving as a greeting) and vertical (radial and ulnar deviation Image 3.2 – like fanning oneself) due to how the controller is held. Occasionally players may incorporate forearm movements to increase their range of motion. Pointing is a continuous action; over extended periods of time the repetitive motions involved may cause fatigue and stress in the user, but we will only consider that in future work.

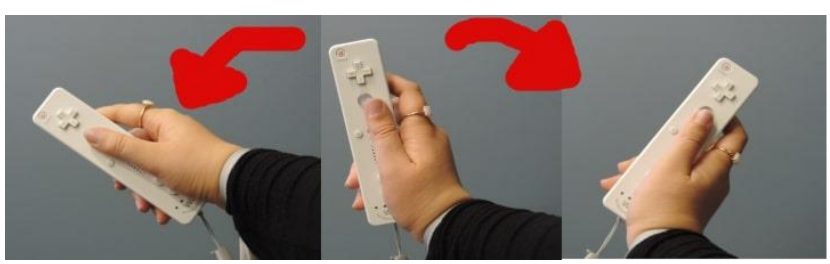


Image 3.2: Wrist Flexion and Extension while holding a handheld motion controller

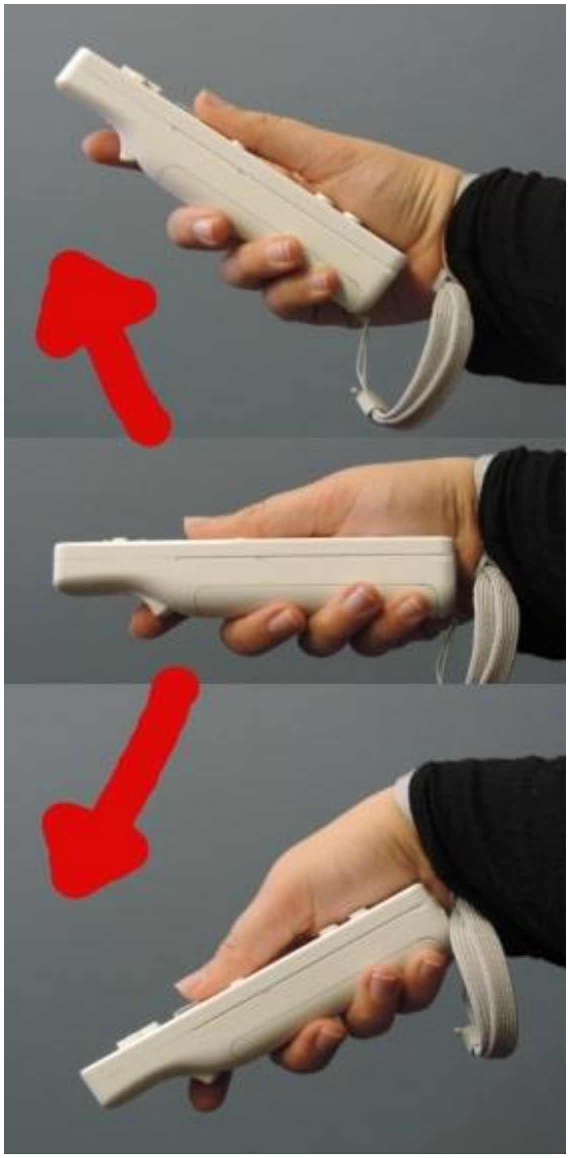


Image 3.3: Wrist Deviation while holding a handheld motion controller

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 affects the completion speeds, and how and 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, as well as 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 Image 3.3). 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 for handheld motion controllers. In comparison, handheld motion controllers held horizontally, smartphones/tablets in landscape mode and handheld consoles are held between the hands; tilting up and down, the motion remains the same as vertical tilting for handheld motion controllers (radial and ulnar deviation). When tilting laterally, the wrist's main function is stability and the tilting motion is performed by coordinated movement of the forearms (forearm flexion and extension Image 3.4). 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), done by tilting the handheld motion controller. Tilting is a continuous action, like pointing, but the additional twisting movement is enough difference to keep them separate.

Image 3.4: Wrist Supination and Pronation while holding a handheld motion controller

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), which act as a brush proxy to paint in the air (canvas); smartphones/tablets, and handheld consoles, where the brush proxy is either a finger or a stylus used to paint on a touchscreen (canvas). Drawing predominantly uses wrist motions, taking advantage of its lateral, vertical, and rotational movements; forearm movements may be incorporated for larger canvases (that surpass wrist range) – which makes it distinctly different from the previous two 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). A minimum of two distinct lateral wrist movement occurs (back and forth), though more can be performed to repeat the in-game actions. The difference between swinging and flicking is their speed; flicking is fast and less precise, while swinging can be steady and accurate. The difference in number of movements and speed distinguishes these actions.

Image 3.5: Forearm flexion and extension while holding a handheld console (Wii U)

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 (forearm rotation Image 3.6) similar to the motion of shaking a cocktail shaker. Examples include ground pound in Donkey Kong Country Returns when using a handheld motion controller (Retro Studios, 2010) and asteroid in SpaceTeam on smartphones/tablets (Sleeping Beast Games, 2012). 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).

Image 3.6: Forearm rotation while holding a handheld motion controller

We were unable to find examples of shaking for landscape smartphones/tablets and handheld consoles. This is possibly because these have the screen attached, so shaking the controls shakes the screen too, making the game extremely difficult to play. The movements for all shaking contexts are sufficiently different to remain separate.

Neck and Face

Head movements such as tilting, nodding and shaking are done by moving the neck. 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 can be seen 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, rather 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

After refinement we have the following motor abilities:

Fine motor abilities	Hands	Fingers	Pressing
			Swiping
			Pinching
		Wrist	Shaking
			Flicking
			Pointing
			Swinging
			Drawing
			Tilting
	Head	Neck	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

Table 3.2: All motor abilities possible to interact with video games.

A generic player homunculus has these and the basic cognitive abilities (attention, perception, memory). We represent it as a bar graph; abilities are listed along the bottom and given as a number between 0 and 100. A score of 0 means they are not able to use that ability. 100 means they completed normative development and can use unencumbered. The generic player homunculus being an abstract representation of an able-bodied normative player has the following profile:

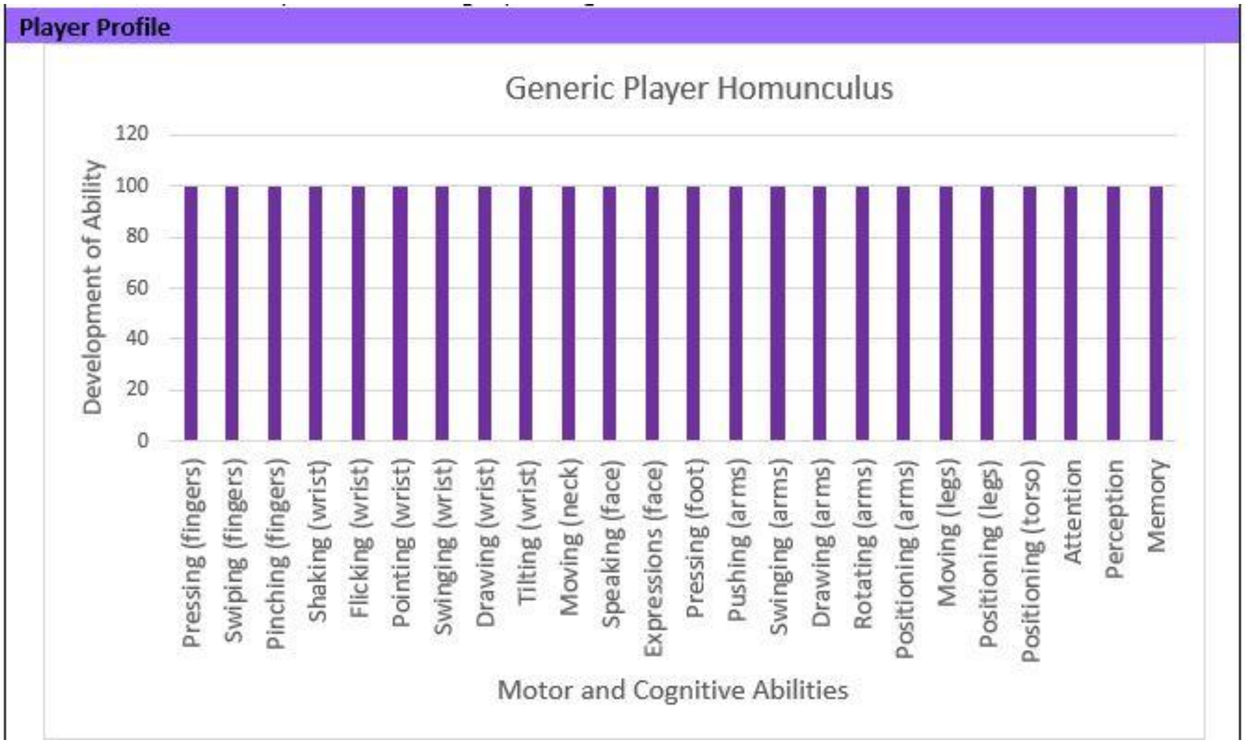


Table 3.3: Generic Player Homunculus

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.

4.0 Gameplay Challenges

We define a *gameplay challenge*[1] *as an 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 adequately described by the set of challenges they use (Adams, 2010; McMahon, Wyeth, & Johnson, 2015; Feil & Scattergood, 2005; Djaouti, Alvarex, Jessel, Methel, & Molinier, 2008; Veli-Matti, 2014). This positions challenges as the unit tasks of gameplay, describing individual tasks the player must accomplish. While this is an overly reductionist approach which doesn't extend to all aspects of the player experience, we can nevertheless adopt it while studying the mechanical experience. Though previous works differ on what makes a challenge, they all define them along the lines of goals and mechanics.

Viewing challenges as tasks, we can characterize them by their *competency profiles* – the set of human abilities (motor, cognitive, emotional, etc.) that are needed to succeed at the task (Fleishman, Quaintance, & Broedling, 1984). This is different from the mechanics, which are

---

[1] Referred to subsequently as just *challenge*.

often 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). The competency profile approximates how much any ability is used in the successful completion of a challenge by assigning a level of use to each ability. By characterizing challenges through their competency profiles, we can get an idea of their intrinsic mechanical experience. The *intrinsic competency profile* informs the designer about the *expected mechanical experience* of the challenge for normative, unencumbered players.

To describe challenges by their competency profiles, we need to have a list of atomic challenges found across games that we can analyse. Several researchers have attempted to produce such lists, defining commonly found types of challenges. To decide if we can use these lists, we need criteria to judge if they have appropriate challenge descriptions. We believe that a good challenge description must delineate between similar challenges, and thus include:

1. the in-game mechanics associated to the challenge,
2. the mechanism of interaction between the player and the game (i.e. the inputs and outputs), and
3. the competency profile.

The first point lets us capture to capture the challenge's goals and context, while the third point is what allows us to distinguish between similar abstract challenges with different player mechanics. As challenge taxonomies tend not to cover the mechanisms of interaction, explicitly being hardware agnostic, none will be sufficient, but can still be a good starting point. However, hardware agnosticism can lead to unintentional blindness to the sources of difficulty. Consider Just Dance on the Nintendo Switch versus on the Xbox One. The gameplay is the same; however, the Switch tracks this motion through the Joy-Con controller (tracking only one arm), while the Xbox One uses the Kinect (tracking the whole body). On the Switch it is possible, for players with mobility issues to play some songs sitting down, which is not possible on the Kinect.

Perspectives on Challenges

We found six frameworks for analysing gameplay and categorizing challenges (Adams, 2010; Djaouti, Alvarex, Jessel, Methel, & Molinier, 2008; Feil & Scattergood, 2005; McMahon, Wyeth, & Johnson, 2015; Veli-Matti, 2014; Bjork & Holopainen, 2004). All six frameworks meet requirement 1 (as this is their purpose), but not 2 or 3.

Karhulahti (Veli-Matti, 2014) defines challenges as a goal with an uncertain outcome (which he borrows from (Malone, 1980)). He proposes two main types: kinesthetic – where the "required nontrivial effort is at least partly psycho-motor"; and non-kinesthetic – where the "required nontrivial effort is entirely cognitive". He defines two types of non-kinesthetic challenges: static (puzzles) and dynamic (strategic). Dynamic is further divided into directly dynamic, indirectly dynamic, totally dynamic, quasi-dynamic, and semi-dynamic. The distinction is based on whether the results of interactions between the system and player are determinate or not. Karhulahti's taxonomy incorporates the idea of mechanical experience by explicitly delineating between the physical and cognitive components of a challenge. He proposes that all gameplay challenges have a non-kinesthetic (mentally solving it) and kinesthetic (executing that solution) component. We agree with this; however, in being concerned with the systems view this framework loses the ability to distinguish between similar but notably different gameplay experiences. Consider the "How Old Is She?" puzzle in Professor Layton and the Diabolical Box (Level-5, 2009) and the trial for "The Turnabout Sisters" case in Phoenix Wright: Ace Attorney (Capcom, 2005), in this framework both are static challenges as each has a specific single path to the correct solution. However, their cognitive experiences are entirely different; for this Professor Layton puzzle players are performing a mathematical equation to arrive at the answer, while the Phoenix Wright trial requires lateral thinking to associate holes in character testimonies to the evidence provided to the player. Therefore, when it comes to distinguishing similar challenges, Karhulahti's taxonomy is insufficient.

Björk and Holopainen outline a framework to describe and analyse games, as well as a set of tools called *patterns* which are "semiformal interdependent descriptions of commonly recurring parts of the design of a game that concerns gameplay" (Bjork & Holopainen, 2004). These patterns are meant to be functional building blocks in game design, and so include elements that can define challenges (e.g. goals, actions, obstacles, mechanics). For our purposes, this framework is limited in that it's solely meant to cover components, not explore the experiences created by these components. While we can combine patterns to create a list of "challenges", we would not be able to distinguish between instances that use the same gameplay components but have different mechanical experiences. Consider the *guard* pattern ("to hinder other players or game elements from accessing a particular area in the game or a particular game element"); when combined with the *characters* ("abstract representations of persons in a game"), *enemies* ("avatars and units that hinder the players trying to complete the goals"), and *aim and shoot* ("the act of taking aim at something and then shooting it") it can describe instances as different as protecting Ashely in Resident Evil 4 (Capcom Production Studio 4, 2005) and protecting Baby Mario in Super Mario World 2: Yoshi's Island (Nintendo EAD, 1995). Though a fitting description of the mechanics of gameplay, it fails to capture the player's experience of these instances (mechanical and otherwise). In Resident Evil 4 the player has limited control over Ashley, being able to command her to hide in safe locations, wait for their movements, or follow the player. In Yoshi's Island Baby Mario is attached to the player and cannot be controlled. Factors such as guarding a character you can control (Ashley) versus a character you can't (Baby Mario) influence how player's approach the problem, which affects the abilities used to address the problem, and therefore the experience profile of the challenge. Similarly, the difference in control mechanisms (joystick versus d-pad) for aiming requires different motor abilities. These differences between similar challenges aren't captured in this type of challenge description. We don't believe we can easily adapt this framework to our needs.

Djaouti et al. created a tool for classifying and analysing gameplay mechanics called "bricks" (Djaouti, Alvarex, Jessel, Methel, & Molinier, 2008). They identified three types of bricks: play, game, and meta. *Play bricks* (manage, random, shoot, write, move, and select) are the in-game actions that a player can take. *Game bricks* (destroy, match, avoid, and create) are the goals of the game. Play bricks and game bricks combine to form *meta-bricks*, which describe families of challenges; e.g. the "DRIVER" meta-brick, which combines the "MOVE" and "AVOID" bricks.

This tool is limited by the scope of games used to create it; they analysed 588 single player “arcade” or “casual” computer games. While the bricks discuss the mechanics of the challenge (as was their original intention) it doesn’t describe the mechanical experience or the influence of the mechanisms of interaction at the level of challenges. Consider the “DRIVER” meta-brick as it represents a family of challenges that require the player move their avatar and avoid obstacles; this describes the main challenge in Need for Speed: Hot Pursuit (Criterion Games, 2010), and Amnesia: The Dark Descent (Frictional Games, 2010). The movement mechanisms in these games is the same (joystick); however, the difference in perspective and goals leads to different underlying experiences. Need for Speed pits the player against the clock, asking them to outperform the other players or NPCs; Amnesia asks the player to run and hide from its stalking monsters. These goals make the problem-solving process underlying them different, as Need for Speed asks players to just focus on the road while Amnesia requires them to be vigilant in exploration and aware of hiding spaces and resources. Due to the difference in context (input mechanisms, camera model, and complexity of gameplay) these experiences are quite different. This framework accurately describes mechanics while unable to capture the experiential differences between instances, making it unsuitable for our purposes.

Feil and Scattergood explain challenges as defined by “objectives, and the barriers that prevent players from achieving [them]” (Feil & Scattergood, 2005), identifying six standard challenges: time, dexterity, endurance, memory/knowledge, cleverness/logic, and resource control. Though these challenges incorporate an element of the player's experience into their definitions, they are too broad; they defined dexterity challenges as “some sort of feat that requires dexterity”. They use both “shooting a target with a pistol” and “[making] quick decisions” as examples of this challenge. Furthermore, they use the term challenge rather liberally – explaining that certain genres put more emphasis on combat, movement, or puzzle challenges without explaining how these relate to the challenges they previously outlined. Nevertheless, their list is closer to what we were looking for in a list of challenges; it attempts to cover the mechanics and hints at the experience but needs significant refinement.

Adams defines challenges as “any task set for the player that is non-trivial to accomplish” (Adams, 2010). He presents 10 major challenge types, subdivided into 30 challenges (see Table 4.1).

Challenge Type	Challenges
Physical Coordination	Speed and Reaction Time Accuracy and Precision Timing and Rhythm Learning Combination Moves
Formal Logic	Deduction and Decoding
Pattern Recognition	Static Patterns Patterns of Movement and Change
Time Pressure	Beating the Clock Achieving something before someone else
Memory and Knowledge	Trivia Recollection of objects or patterns
Exploration Challenges	Identifying spatial relationships Finding keys (unlocking any space) Finding hidden passages Mazes and Illogical spaces
Conflict	Strategy, tactics, and logistics Survival Reduction of enemy forces Defending vulnerable items or units Stealth
Economic	Accumulating resources or points (growth) Establishing efficient production systems Achieving balance or stability in a system Caring for living things
Conceptual Reasoning	Sifting clues from red herrings Detecting hidden meanings Understanding social relationships Lateral thinking
Creation and Construction	Aesthetic success(beauty or elegance) Construction with a functional goal

Table 4.1: Adam's Gameplay Challenges

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 defines rhythm challenges as “tests of the player’s ability to press the right button at the right time” directly referencing the mechanism of interaction and the mechanics. From this definition we can tell that these challenges emphasize motor abilities, justifying it falling under the *Physical Coordination* type. This description of challenges and taxonomy most closely matches our goal. However, its categorization is too broad; instances with different play experiences are joined together. For example, *Speed and Reaction Time* challenges can describe gameplay instances ranging from quick time events to button mashing mini-games. This list is the best starting point we could find but needs refinement.

McMahon et al. revise Adams’ challenges, scoped down to 16 challenges (McMahon, Wyeth, & Johnson, 2015). Through a focus group session, they renamed several of Adams’ challenges and added three new categories. This condensed list confounds challenges; the “thinking outside the box” challenge is associated with the games Portal (Valve Corporation, 2007) and World of Goo (2D Boy, 2008). However, the differences in interaction methods means that the player experiences of these challenges are quite different.

Refining Adams’ List of Challenges

Adams’ list is grounded in gameplay examples and reflects our understanding of challenges being physical and cognitive. As it is closest to our needs, we will start from that taxonomy and refine it. However, it fails to provide crisp definitions for each category, leading to different experiences being lumped together. By creating definitions based on competency profile required for successful completion, we can distinguish different gameplay instances. Nevertheless, our list is still not comprehensive, but rather a starting point for further, testable refinement.

Refinement Methodology

Adams’ definitions are presented inconsistently, 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 both 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 have consistent definition and examples, we find other gameplay instances that fit those descriptions. We attempted to find as many as possible across various “genres” and systems. The purpose of these examples is 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 relies on the subjective knowledge of the researcher (and, to a limited extent, lab peers) to come up with examples; as the work on challenges from this perspective is limited, there is 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 part of the process would benefit from a larger pool of researchers with different gameplaying experiences.

We sort our examples by their *mechanisms of interaction,* as this is the most easily identified difference. We do this by examining the game mechanics and instructions for each instance and the controller used. This first separation accounts 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 accomplished is broadly different ways (pressing buttons, hitting a drum, and swinging your arms). We then use close reading techniques to construct the competency profiles, first identifying required motor abilities (see previous section for list of abilities) then compare with the generic player homunculi in order to find the normative mechanical experience. Difficulty has been noted be an obstacle in close reading as it can pull readers (players) out of the experience (Bizzochi & Tanenbaum, 2011), however since what we're trying to read is the difficulty we believe this won't be a significant barrier.

For each category, we try to add more examples that fit the specific description of the challenge. We separate instances based on game mechanic variants, capturing distinctions like the pushing one or two buttons, or having time limits on the challenge. The purpose is to capture differences in experience due to increased use of attention for coordinated movements, or the use of perception. This allows us to understand the broad cognitive abilities of the competency profile – giving us the rough shape of its competency profile. Then we can rank the competency profile abilities as: not used, used but not noticeably, noticeably used, important but not limiting, or limiting ability.

We repeat this process (add more examples, find distinctions, etc.) until it stabilizes, which tends to happen quickly. We have not had to split more than twice. We then examine 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 is to see whether differences in context creates differences in the motor or cognitive abilities used, leading to further refinement when that is the case.

We deem two challenge instances to be identical if they involve the same motor and cognitive skills from a player, occur over similar periods of time, and are performance bounded by the same skill. We then take re-examine our observations of the examples to assign values to each ability in the competency profile. We assign 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 helps us understand the relationship between abilities in the same category or on the borders. While this remains subjective, we have 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 include examples of multi-player games, we examine them when playing with or against humans, or non-player characters.

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

Name	
Challenge Information	
Definition	Natural language definition of challenge
Mechanics	
	Description of mechanics
Variable components	• Bullet list of mechanics that can be tweaked
Context	
Mechanism of Interaction	
Controller Type	Standard, Handheld Console, Smartphone (portrait), Smartphone (landscape), Handheld Motion Controller (portrait), Handheld Motion Controller (landscape), Full Body Motion Controller, Keyboard, Mouse
Number of Players	Single player, Multi-player
Type of Play	Competitive (solo), Competitive (team based), Cooperative (solo), Cooperative (team based)
Examples	
• Bullet list of game examples	
Variants	
• Bullet list of variant names	
Intrinsic Competency Profile	
	[picture graph of competency profile]
Source of Difficulty	

Table 4.2: Tabular representation of challenge definitions.

We apply this process to Adams' *Speed and Reaction Time* challenges, which we previously noted as incorrectly lumped together. We thus split them and analyse 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). Thus, these challenges contain 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 a particular 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 that *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 then 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 which 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 gives 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. The results of decomposing the remaining Speed Challenges can be found in Appendix A.

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 the 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. Expand outside Nintendo and party games is made more difficult as instances tend to be embedded in larger gameplay segments. Here are 4 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. Manic Mallets has the player hitting a single button as many times as possible in the time limit; Speed Skating requires the player alternate between two buttons; Mecha-Marathon requires pressing 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 in used abilities divides 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.

Manic Mallets has teams of 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 aren'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) realised 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 have a partner whose contribution is meaningful put less pressure on us to perform optimally. This change in our feeling of pressure does not affect the competency profile as it didn't change our approach to the gameplay. Playing 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 was 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 identify *finger pressing, attention, and perception* as the three abilities in this competency profile.

Dragon's Breath 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 is 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's 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.

Image 4.1: Stick of Truth's Dragon's Breath, single input button mashing with implicit time limit

Torture attacks in both Bayonetta and Bayonetta 2 are a triggered combat action which removes the player from regular combat[2] 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 does 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.

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 approximate the level of use of finger pressing to be 90 – making it the limiting ability. We use a value of 90 to communicate that we're confident that testing will reveal as the limiting ability, but conscious of potential error. We find that attention and perception are both used, but not noticeably enough so as to imperil success. From play, the importance of these abilities is 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 assign attention a 15: used, but not important. With the margin of error of our estimates, we allow that in sufficiently stressful

[2] They cannot change targets or be hit with other attacks for the duration of the torture attack.

circumstances attention may cross over into noticeably used territory. Perception seemed to have minimal effect on our performance. Beyond awareness that we are pressing the right button, we could play these instances blindfolded if given a cue to start. Thus we assign perception a 10, leaving it squarely in the used, but not important category, even with the margin of error.

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 4.3: Tabular representation of single input button mashing challenges.

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've found have the same goals, mechanics, and mechanisms of interaction and thus does not need further decomposition. We analyse these examples for their competency profile.

Mecha-Marathon has each player competing against the others to wind up a doll by simultaneously mashing the A and B button (on a standard controller) within a ten 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 isn'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, but this time holding the controller in one hand and resting our thumb across the A and B buttons, pressing being done as a movement of the base knuckle of the thumb (near the palm), we found this position more comfortable, but it didn't improve our performance against the NPCs, 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.

Image 4.2: Mecha-Marathon instructions and gameplay

Chin Up Champ has players compete at performing the most chin-ups in ten 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 is 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 (Image 4.3).

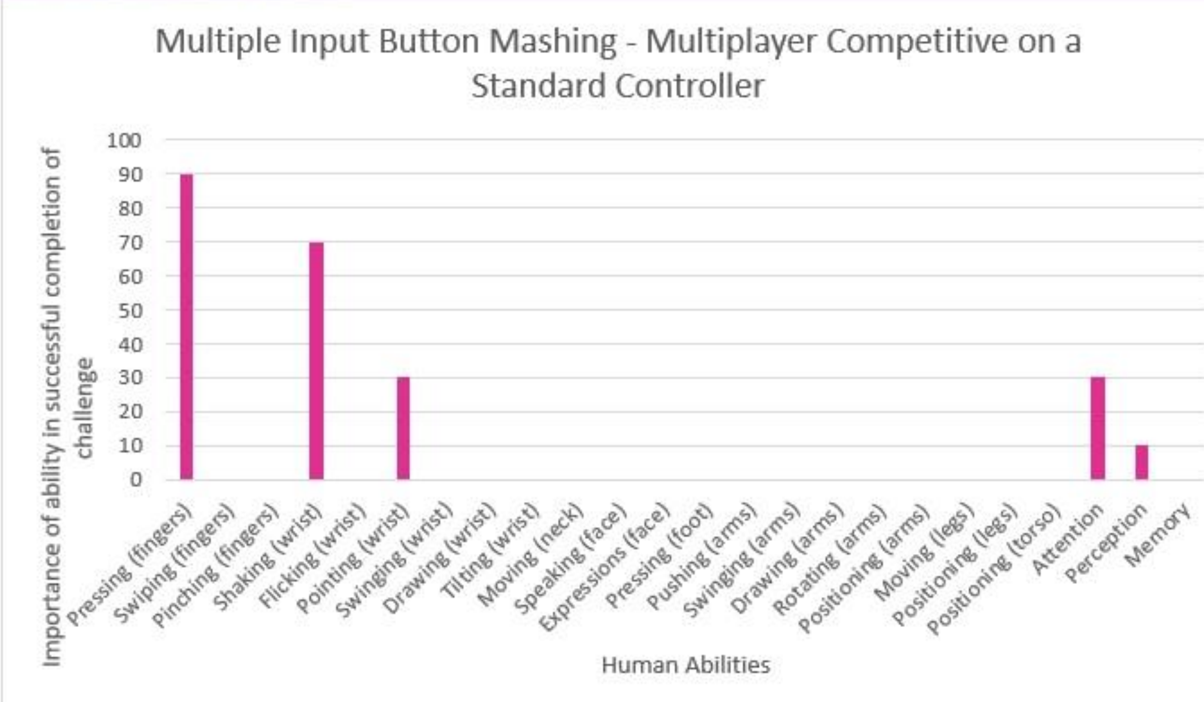
Image 4.3: Comparison of A and B button placements between Wii Remote and N64 controller.

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 of this challenge. 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 this challenge 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, so that designers do not consider them to be different enough, and thus choose to use the cognitively simpler single input button mashing instead.

In Table 4.4, we show the competency profile (though warn that the paucity of examples means there is more room for error). Unlike SIBM, the competency profile does vary depending on the example. In both examples, 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 approximate 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. However, it was never taxing, merely noticeably used (30). Attention is like wrist pointing. noticeably used (30) in all scenarios, although for different reasons. Perception on the other hand we find to have minimal effect on our performance, assigning it a 10 (used, not important, and less so that the others). Since wrist/forearm shaking only appears in a single variant (high pressure standard controller contexts) it is certainly not a limiting ability of the abstract challenge type. However, it significantly affects our performance against human players (in Mecha-Marathon). We give it a 70, as important but not limiting. Given our built-in margin of error, this documents the importance of this ability, while leaving room for it to cross into the territory of limiting ability as difficulty rises.

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

Table 4.4: Tabular representation of multiple input button mashing challenges.

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 3). Adams included the Track and Field example here, but for brevity we will skip it. This grouping has converged on mechanics, goals, and mechanisms of interaction so we analyse their competency profiles.

Psychic Safari tasks two players to power up an ancient relic to destroy their opponent's relic. There is an explicit 5 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 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.

Image 4.4: Psychic Safari gameplay.

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 5 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 coordinate across both hands. We don't believe this affects the amount of attentional resources needed as we're not coordinating our movements to be simultaneous, but just so that they happen 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 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 3 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 set up felt more natural, as it allowed to hold the controller in a reasonable way while leaving our fingers resting on both buttons.

Balloon Burst 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 (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 3 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 wasn't as important. Where previously we'd be mashing button 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.

Image 4.5: Balloon Burst instructions and gameplay from Mario Party 2

Ridiculous Relay is a race between a solo player and a three-player group; it has two types of player experiences. Here we will be 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 controls the right and left oar. The time limit is implicit as the player needs to just move fast enough to cover the distance of their segment of the race. When playing this we realised that the 3-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 didn’t concern ourselves with the performance of others because we couldn’t influence the result. We found in the shell section that the abilities we used seemed the same as previous examples: finger pressing, wrist pointing, attention, perception, and memory.

Speed Skating is a race between the player and 3 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 combat related). 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.

In Table 4.5, we show the competency profile. Though the contexts of the examples differed, we found the abilities used and their amounts were fairly 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 feel was slightly more important than wrist pointing (as an incorrect button press could cost us the challenge) so we list 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 rather it as a (20).

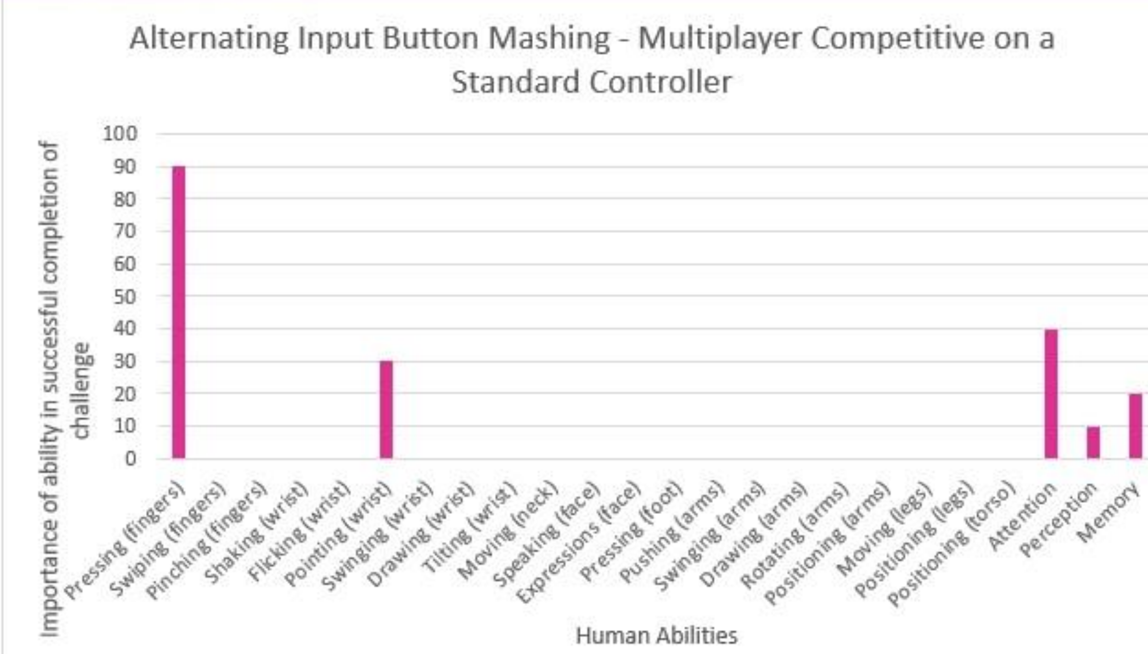
Name	Alternating Input Button Mashing
Challenge Information	
Definition	Repeatedly and rapidly press two specific buttons in sequence.
Mechanics	
Button pressing, short time limits, pattern recognition	
Variable components	• Length of sequence/number of buttons pressed • Which buttons are being pressed • 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

Table 4.5: Tabular representation of Alternate Input Button Mashing challenge

5.0 Challenge Jutsus: A Knowledge Capture Artifact

One of our goals is to develop an adequate 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 jutsus* to link them together.

As a knowledge capture mechanism, challenge jutsus 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, but with a different focus: design patterns are solution-focused while jutsus are about framing the problem and its root causes. We see design patterns as emerging from challenge jutsus once context of occurrence is fixed.

5.1 Challenge Jutsu Methodology

We document challenge jutsus 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 particular demographic of targeted players. It is represented as a bar graph, with each bar representing proficiency in an associated ability compared to a normative adult. Here we will only discuss generic players, i.e. those the abilities of a normative adult, as previously described.

The derived mechanical experience relates the challenge and player descriptions. It is represented with 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) are visible and recorded in the jutsu. Identified barriers to challenge completion enable us to offer a shortlist of tweaks that could be made to the variable components of the challenge to address them.

Finally, a challenge jutsu is represented in a tabular form:

Name	
Challenge Information	
Definition	Natural language definition of challenge
Mechanics	
	Description of mechanics
Variable components	• Bullet list of mechanics that can be tweaked
Context	
Mechanism of Interaction	
Controller Type	Standard, Handheld Console, Smartphone (portrait), Smartphone (landscape), Handheld Motion Controller (portrait), Handheld Motion Controller (landscape), Full Body Motion Controller, Keyboard, Mouse
Number of Players	Single player, Multi-player
Type of Play	Competitive (solo), Competitive (team based), Cooperative (solo), Cooperative (team based)
Examples	
• Bullet list of game examples	
Variants	
• Bullet list of variant names	
Intrinsic Competency Profile	
	[picture graph of competency profile]
Source of Difficulty	
Player Profile	
	[picture graph of player profile]
Mechanical Experience	
[overlapped competency and player profile with problem areas highlighted]	
Areas of difficulty	• Bullet list of abilities from competency profile that are higher than the player's abilities
Suggested Tweaks	• Bullet list of how to tweak variable components to address the areas of difficulty

Table 5.1: Template of Challenge Jutsu

5.2 Challenge Jutsu: Single Input Button Mashing

We apply the above to Single Input Button Mashing challenges to detail its challenge jutsu. The challenge description is exactly that from Section 4.3.1.1 (Table 4.3).

The player description, as we mentioned already, will be taken as the Generic Player Homunculus from Section 3.1.3 (Table 3.3).

Then we compare the competency profile with the player's abilities. When the player's ability is significantly lower, we highlight the difference in red. Red thus indicates 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 can then attempt 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 highlight the ability in yellow. Yellow abilities may not affect the player's chances of success.

Player’s may compensate for yellow abilities by using supporting abilities, or alternatively can train these abilities for these contexts. We call these identified areas *potential sources of difficulty* as their effect on play varies from player to player.

Note that 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. Using the Generic Player Homunculus, we have the following “actual” mechanical experience:

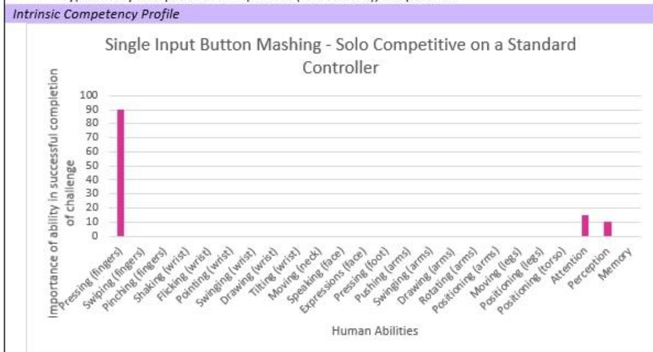
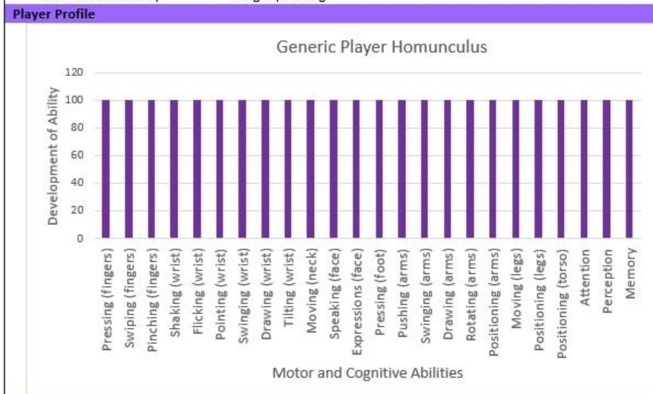
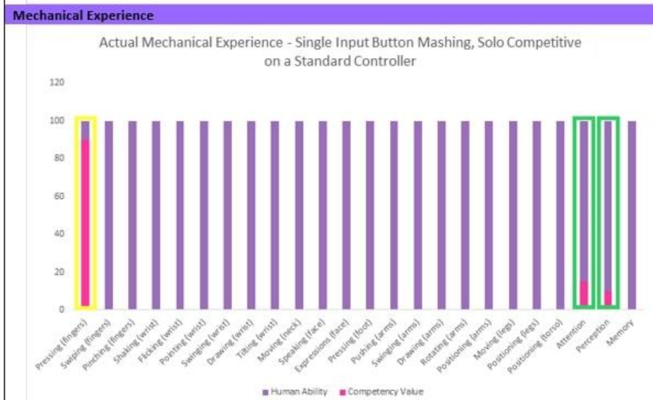
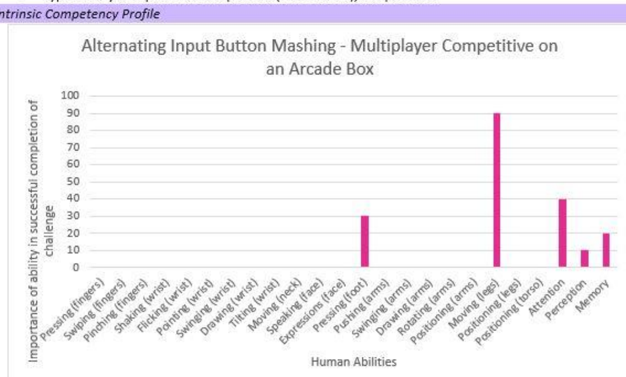
Name	Single Input Button Mashing
Challenge Information	
Definition	Repeatedly pressing a specific single button or key on the controller as fast as possible within a given time limit.
Mechanics	
Button pressing, short time limits	
Variable components	• Time limit
Context	
Mechanism of Interaction	Pressing 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
Player Profile	
	
Mechanical Experience	
	
Areas of difficulty	• N/A
Suggested Tweaks	• N/A

Table 5.2: Challenge Jutsu of Single Input Button Mashing on a Standard Controller for Generic Player Homunculus

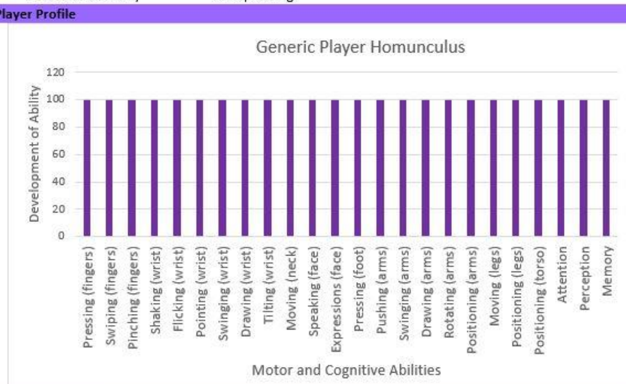
5.3 Variants in Challenge Jutsu

Challenge jutsus make it easier to document and compare similar challenges. 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 needlessly multiplying jutsus (the mechanism of interaction is different after all), we instead use *jutsu variants*. *Jutsu variants* are used when the same challenge exists across different motor abilities; for example, substituting finger pressing with foot pressing or arm pressing. The jutsu variants don't 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.

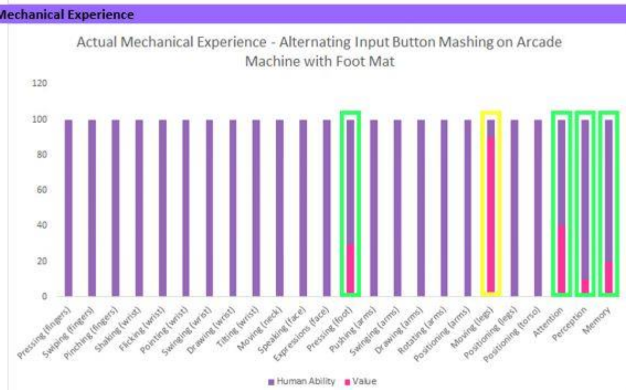
Name	Single Input Button Mashing
Challenge Information	
Definition	Repeatedly pressing a specific single button or key on the controller as fast as possible within a given time limit.
Mechanics	
Button pressing, short time limits	
Variable components	• Time limit
Context	
Mechanism of Interaction	Pressing buttons
Controller Type	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
Player Profile	



Mechanical Experience	



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

Table 5.3: Jutsu Variant of Alternating Input Button Mashing on an Arcade Box for Generic Player Homunculus

5.4 Collection and Organization of Challenge Jutsu

An organized system of jutsus 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 thus propose an online, public database. This way, via different views, we can accommodate both user groups: researchers looking to analyse existing games, and designers looking to create new games. For ease of use, a structured wiki as a front-end is our first choice.

Organizing the numerous jutsus 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.

The primary analysis view currently sorts the jutsus by their source of difficulty (Image 5.1), as this fits our own research agenda. When other dimensions of game analysis that fit the jutsu pattern become clear, we can add such views. The analysis view is very useful for understanding interaction barriers (and thus accessibility problems). For example, investigating whether a game is playable by children with cerebral palsy (maybe as part of a larger study), 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.

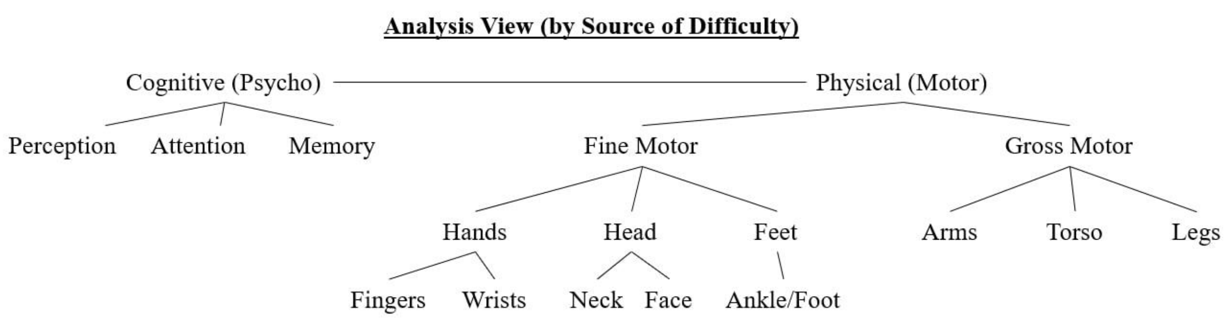


Image 5.1: Analysis-View organization of Challenge Jutsu Database

The design view sorts the jutsus by their challenge types (Image 5.2). 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 to them ability information.

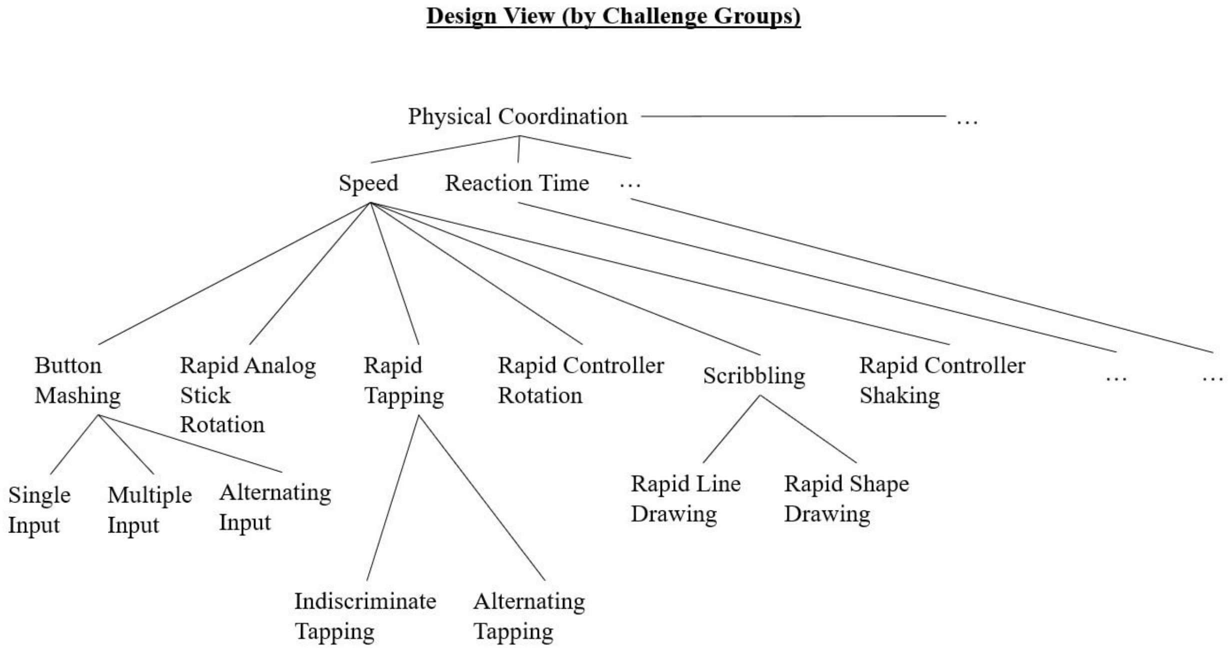


Image 5.2: Design-View organization of Challenge Jutsu Database

6.0 Discussion and Conclusion

We have defined a new framework to analyse and discuss the mechanical player experience, by detailing a methodology for refining gameplay challenges, a generic player homunculus, and the organizational structures for their comparison (challenge jutsus, and the challenge jutsu database). We outline some potential applications and benefits of this work, as well as what further refinements could be made to the data and methodology.

6.1. Potential Applications

The first use for our framework 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 profile and specific demographic abilities can drive the selection of test cases, hopefully leading to reduced testing time around playability.

The framework can also be used to systematically explore 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 a portion of the demographic and not others. This 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, this also allows for the exploration of novel challenges. As gaming continues to grow as an industry, novelty becomes more difficult to achieve and more important as a selling feature. Similarly, we can use it to explore why challenges only work in specific contexts, helping to understand how to adapt existing challenges to create new jutsu variants. The structured presentation of the jutsus also allows us to see which abilities are underrepresented or even unused in challenges. This lets us create new challenges and gives designers a guide to explore that space.

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.

6.2. Future Work and Conclusions

We provide a rationale for, and methodology for constructing, challenge jutsu. We plan to work on further refinements, and then experimental validation. Expanding the work to encompass Adams' 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.

Understanding the mechanical experience of game challenges is a necessary first step in dealing with accessibility issues. We start that work by presenting framework and methodology for that, as well as understanding the sources of difficulty for challenges. We showed this framework in action by illustrating it on subset of Speed challenges. We explained how to capture and structure this information into individual jutsu and how to organize these jutsus into a database. We outlined how this structured presentation of information helps through making certain information more noticeable. This work is ready to be expanded – 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 in ways we couldn't before. We want to help improve the player experience for many under-served gamer demographics.

Bibliography

2D Boy. (2008, October 13). World of Goo. *Windows*. Redmond, Washington, United States of America: Microsoft Game Studios.

Adams, E. (2010). *Fundamentals of Game Design* (2nd ed.). Berkley, California, United States of America: New Riders.

Atlus. (2005, October 4). Trauma Center: Under the Knife. *DS*. Irvine, California, United States of America: Atlus USA.

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.

Bizzocchi, J., & Tanaenbaum, J. (2011). Well read: Applying close reading techniques to gameplay experiences. *Well Played 3.0: Video Games, Value and Meaning*, 262-290.

Bjork, S., & Holopainen, J. (2004). *Patterns in Game Design.* Needham Heights, Massacheussets, United States of America: Charles River Media.

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

Capcom. (2005, October 12). Phoenix Wright: Ace Attorney. Chuoku, Osaka, Japan: Capcom.

Capcom. (2014, August 29). Professor Layton vs. Phoenix Wright Ace Attorney. Kyoto, Kansai, Japan: Nintendo.

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

Capcom Production Studio 4. (2005, January 11). Resident Evil 4. *Gamecube*. Osaka, Japan: Capcom.

Card, S. K., Moran, T. P., & Newell, A. (1983). *The Psychology of Human-Computer Interaction.* Hillsdale, New Jersey, United States of America: L. Erlbaum Associates Inc.

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

Chen, J. (2007). Flow in games (and everything else). *ACM. 4*, pp. 31-34. ACM. doi:10.1145/1232743.1232769

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

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

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

Criterion Games. (2010, November 26). Need for Speed: Hot Pursuit. *Xbox 360*. Redwood City, California, United States of America: Electronic Arts.

Csikszentmihalyi, M. (1997). *Finding flow: The psychology of engagement with everyday life.* New York, New York, United States of America: Basic Books.

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

Dillon, R. (2012). Teaching Games Through the AGE Framework. *IGDA Perspectives Newsletter*, pp. 1-7.

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.

Epyx. (1985). Summer Games II. *Commodore 64*. Pennsylvania, United States of America: Commodore Gaming.

Feil, J., & Scattergood, M. (2005). *Beginning game level design.* Boston, Massacheussets, United States of America: Thomson Course Technology.

Firaxis Games. (2010, September 21). Sid Meier's Civilization V. Novato, California, United States of America: 2K Games.

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

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

FromSoftware. (2019, March 22). Sekiro: Shadows Die Twice. Santa Monica, California, United States of America: Activision.

Frost, J., & Baird, S. (2013, September 13). Dumb Ways to Die. *Android*. Ontario, Canada: PopReach, and Metro Trains Melbourne.

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

Game Freak. (2013, October 12). Pokemon X & Y. *3DS*. Kyoto, Kyoto, Japan: Nintendo.

Gamma, E., Helm, R., Johnson, R., & Vlissides, J. (1994). *Design Patterns: Elements of Reusable Object-Oriented Software.* United States of America: 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.

HAL Laboratory. (2002, December 2). Kirby: Nightmare in Dreamland! *Gameboy Advance*. Kyoto, Kyoto, Japan: Nintendo.

HAL Laboratory, & Nintendo R&D 2. (2001, April 11). Kirby Tilt 'n' Tumble. *Gameboy Colour*. Kyoto, Kyoto, Japan: Nintendo.

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*, pp. 17-22.

id Software. (1996, June 22). Quake. *Windows*. New York City, New York, United States of America: GT Interactive.

Intelligent Systems, & Nintendo SPD. (2005, February 14). WarioWare: Touched! *DS*. Kyoto, Japan: Nintendo.

Intelligent Systems, & Nintendo SPD. (2007, January 15). WarioWare: Smooth Moves. *Wii*. Kyoto, Japan: Nintendo.

Intelligent Systems, & Nintendo SPD Group No. 1. (2005, May 23). WarioWare: Twisted. *Gameboy Advance*. Kyoto, Japan: Nintendo.

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

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

Konami Computer Entertainment Japan. (2001, November 12). Metal Gear Solid 2: Sons of Liberty. *Playstation 2*. Tokyo, Japan: Konami.

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

Lindenberger, U., Marsiske, M., & Baltes, P. B. (2000). Memorizing while walking: increase in dual-task costs from young adulthood to old age. *Psychology and aging, 15*(3), 417.

Mackenzie, I. S. (2013). *Human-computer interaction: An empirical research perspective.* Waltham, Massacheussetts, United States of America: Mogran Kaufmann.

Malcolm, B., Foxe, J., Butler, J., & De Sanctis, P. (2015). The adging brain shows less flexible reallocation of cognitive resources during tual-task walking: A mobile brain/body imaging (MoBI). *NeuroImage*, *117*, 230.

Malone, T. (1980). What Makes Things Fun to Learn? Heuristics for Designing Instructional Computer Games. *Proceedings of the 3rd ACM SIGSMALL Symposium and the First SIGPC Symposium on Small Systems* (pp. 162-169). Palo Alto: ACM. doi:10.1145/800088.802839

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

Metro Trains Melbourne. (2018, January 23). Dumb Ways to Die 2. *Android*. Ontario, Canada: PopReach.

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

Naughty Dog. (2013, June 14). The Last of Us. *Playstation 4*. California, United States of America: Sony Computer Entertainment.

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

Nintendo EAD. (1995, October). Super Mario World 2: Yoshi's Island. *SNES*. 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.

Office Create Corporation. (2018, May 31). Cooking Mama: Let's Cook! *Android*. Kanagawa, Japan: Office Create Corporation.

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.

Piaget, J. (1983). Piaget's Theory. *Handbook of Child Psychology, 1*.

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, United States of America: 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). Mario 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). Mario 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.

Snap Inc. (2018, April 25). Snapchat Snappables. *Android*. California, United States of America: Snap Inc.

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.

Square Product and Development Division 1. (2001, December 17). Final Fantasy X. *PlayStation 2*. California, United States of America: Square Electronic Arts.

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

Sweester, P., & Wyeth, P. (2005). Gameflow: A model for evaluating player enjoyment in games. *ACM Computers in Entertainment*, (pp. 3-3).

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.

Ubisoft Paris, Ubisoft Pune, Ubisoft Shanghai. (2018, October 23). Just Dance 2019. *Nintendo Switch*. Rennes, France: Ubisoft.

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

Valve Corporation. (2007, October 10). Portal. *Windows*. Bellevue, Washington, United States of America: Valve Corporation.

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

Walk, W., Gorlich, D., & Barrett, M. (2017). Design, Dynamics, Experience (DDE): An Advancement of the MDA Framework for Game Design. *Game Dynamics*, 27-45.

Yee, N. (2007). Motivations of Play in Online Games. *Journal of CyberPsychology and Behaviour*, *9*, pp. 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*, pp. 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*, pp. 1-7. New York: ACM. doi:10.1145/3235765.3235804