Types for Players

Jacques Carette, Sasha Soraine
McMaster University
{carette,sorainsm}@mcmaster.ca

Games are made up of different kinds of experience; game designers are *experience designers* for players. We are specifically interested in the *mechanical experiences* of players interacting with challenges.

Impossible challenges for any human's abilities are trivial to create. One class of "mechanically interesting challenges" is one calibrated so that the **capabilities of the player** are noticeably stressed, but not to their breaking point. This view has been a focus of flow (e.g. (Csikszentmihalyi 1997; Sweetser and Wyeth 2005)), and dynamic difficulty adjustment (DDA) (e.g (Denisova, Guckelsberger, and Zendle 2017; Zohaib and Nakanishi 2018)). We want to better understand player capabilities to understand how to create more tailored experiences.

A standard Human-Computer Interaction (HCI) view sees the running trace of a game as a sequence of communications between two 'programs', the player and the game (Fig. 1). We would like to see both sides as *typed*, and their interaction as *typed* too.

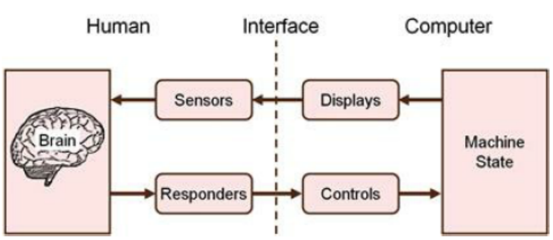

Figure 1: Abstract view of interaction from HCI.

But does it even make sense to assign 'types' to a player? Work on "player types" is abundant; the focus of player typologies have been behavioural (e.g. (Bartle 1996; Yee 2006)), or psychometric (e.g. (Tseng and Teng 2015; Zackariasson, Wåhlin, and Wilson 2010)) divisions. There are many criticisms regarding the usefulness and validity of these typologies (e.g. (Bateman et al. 2011)), particularly around boxing players into categories.

We have a clash of terminology: we mean *type* as used in programming languages, not in game studies! Let us say instead that a type for X is a description of a static property of X. More specifically, *the type of a player is the set of constraints expressed by that player's capabilities – physical and cognitive*. Constraints like how fast can the player react (correctly) to seeing a particular event happen on screen; or how fast can a choice of appropriate reaction be made.

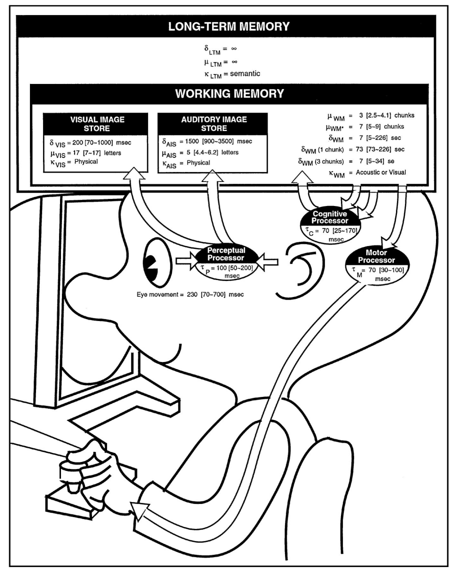

Figure 2: The Model Human Processor(Card, Newell, and Moran 1983), giving processor speeds for various processes.

Types do not exist in a vacuum, they need to assemble into a coherent, compositional *type system* to be useful. The predicates that interest us describe ability constraints that are measurable and correlate with being able to conquer game challenges.

Which gives us a starting set of questions:

1. what capabilities?
2. what measures?
3. are the capabilities and measures adequate?
4. is this an ethical line of research?

The Model Human Processor (Card, Newell, and Moran 1983) is an example of modelling a user as constrained abilities based around measured ability limitations such as the rate of information decay in memory stores (Fig. 2). But we'd like more precise abilities than just generic perceptual, cognitive and/or motor processing limitations.

We can critically examine some game examples as a means of synthesizing applicable capabilities. Consider **Messy Memory** from Mario Party 3 (Hudson Soft 2001). Players have 3 seconds to remember the positions of 10 items before they are scrambled (Fig. 3a). Players are then given 10 seconds to put them back in the correct order (Fig. 3b), with the player closest to the correct sequence winning (games can end in multiple winners). Obviously short-term memory is crucial. The primary constraint is a player's working set size of their short-term memory and, secondarily, how fast they can move items into their proper place.

(a) (b)

Figure 3: Messy Memory from Mario Party 3.

Or **Looking for Love** in Super Mario Party (Nd Cube 2018), where the player must be first to look at the heart when it appears. This game is a *a choice reaction task*, relying on recognition, planning (to pick the right direction) and motor execution. The task is made harder by the game presenting more options (the other suits, Fig. 4b), or trying to trick the player's senses by changing the colour of the heart or of the other wrong choices — messing with the player's instinct to go after "red" (Fig. 4a). Thus the game actually tests for *inhibition*, *object recognition*, and *selective attention/reaction time*.

(a) (b)

Figure 4: Looking for Love from Super Mario Party.

Both examples also involve motor constraints: using the controller correctly and quickly. At this level of psychomotor, events happen in a matter of milliseconds. Bailey's work on the time frame of reaction tasks (Bailey 1996) gives us an idea of how long these take generically (Tbl. 1).

So what would type rules look like? For generic humans, we could have

$$\frac{}{\texttt{sense} : [1, 38]} \qquad \frac{}{\texttt{process} : [70, 300]}$$

Operation	Typical Time (ms)
Sensory reception	1 — 38
Neural transmission to brain	2 — 100
Cognitive procession	70 — 300
Neural transmission to muscle	10 — 20
Muscle latency and activation	30 — 70
Total	113 — 528

Table 1: Measures of cognitive operation in reaction time.

$$\frac{a : [t_1, t_2] \quad b : [t_3, t_4] \quad c : {}_a||_b\, t_5}{a; b : [t_1 + t_2 - t_5, t_2 + t_4]}$$

where $[-, -]$ denotes a time interval in milliseconds, ${}_a||_b\, t$ that processes a and b may be done in parallel "up to" t milliseconds, and we use ; for sequential composition. There is a rich literature around reactive systems (Wan and Hudak 2000; Jeffrey 2012) with novel ideas that could likely be adapted for such type systems (Bahr, Graulund, and Møgelberg 2021; Graulund, Szamozvancev, and Krishnaswami 2021).

More interesting would be to *measure* detailed, specific operations on a per-player basis, which would significantly narrow the given intervals. Players would have to *consent* to such measures, and games would have to be taught to adapt accordingly. Note that games currently do this without consent (via DDA), albeit on fairly crude inputs.

Player type systems open up the possibility of static type checking for experiences. If the designer knows what kind of experience they are trying to create, ability models can let one verify that a challenge is *achievable*. More precisely, if a player's process for beating a challenge c is p, then if that process can take less time that is allotted for the challenge, that challenge is *achievable*.

$$\frac{p : [t_1, t_2] \quad c : [0, t_3] \quad t_1 < t_3}{c\ \texttt{achievable}}$$

If t_1 and t_3 are extremely close, then the player might be extremely frustrated, as "peak performance" is stressful.

In other words, rather than a seeing these time intervals as purely what is feasible, a more refined model would use a probability distribution and a convolution would be needed to perform sequencing. A still more refined model could introduce stress, fatigue and other similar factors.

It also opens up the possibility of recontextualizing DDA as dynamic type checking on player types; since DDA aims to align the game with the player's abilities, in essence it could be seen as type checking.

There are a whole host of open questions:

- is this really compositional? (|| is iffy)
- does it scale to longer time frames?
- does it scale to other parts of the player experience?
- how many pieces of the model of Fig. 1 should be seen as independent components?
- is such data collection ethical?
- would this reinforce ableism or highlight unconscious assumptions at design time, when they are easier to fix?

References

Bahr, P.; Graulund, C. U.; and Møgelberg, R. E. 2021. Diamonds are not forever: liveness in reactive programming with guarded recursion. *Proceedings of the ACM on Programming Languages* 5(POPL):1–28.

Bailey, R. W. 1996. *Human performance engineering designing high quality professional user interfaces for computer products, applications and systems*. Upper Saddle River, New Jersey, USA: Prentice-Hall, Inc.

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

Bateman, C.; Lowenhaupt, R.; Nacke, L. E.; et al. 2011. Player typology in theory and practice. In *DiGRA Conference*.

Card, S. K.; Newell, A.; and Moran, T. P. 1983. *The Psychology of Human-Computer Interaction*. Hillsdale, NJ, USA: L. Erlbaum Associates Inc.

Csikszentmihalyi, M. 1997. *Finding flow: The psychology of engagement with everyday life.* New York, NY, USA: Basic Books.

Denisova, A.; Guckelsberger, C.; and Zendle, D. 2017. Challenge in digital games: Towards developing a measurement tool. In *Proceedings of the 2017 chi conference extended abstracts on human factors in computing systems*, 2511–2519.

Graulund, C. U.; Szamozvancev, D.; and Krishnaswami, N. 2021. Adjoint reactive gui programming. In *FoSSaCS*, 289–309.

Hudson Soft. 2001. *Mario Party 3*. Game [N64]. Nintendo, Kyoto, Japan.

Jeffrey, A. 2012. Ltl types frp: linear-time temporal logic propositions as types, proofs as functional reactive programs. In *Proceedings of the sixth workshop on Programming languages meets program verification*, 49–60.

Nd Cube. 2018. *Super Mario Party*. Game [Switch]. Nintendo, Kyoto, Japan.

Sweetser, P., and Wyeth, P. 2005. Gameflow: A model for evaluating player enjoyment in games. *Comput. Entertain.* 3(3):3–3.

Tseng, F.-C., and Teng, C.-I. 2015. Online gamers' preferences for online game charging mechanisms: The effect of exploration motivation. *Int. J. E-Bus. Res.* 11(1):23–34.

Wan, Z., and Hudak, P. 2000. Functional reactive programming from first principles. In *Proceedings of the ACM SIGPLAN 2000 conference on Programming language design and implementation*, 242–252.

Yee, N. 2006. Motivations for play in online games. *CyberPsychology & behavior* 9(6):772–775.

Zackariasson, P.; Wåhlin, N.; and Wilson, T. L. 2010. Virtual identities and market segmentation in marketing in and through massively multiplayer online games (mmogs). *Services Marketing Quarterly* 31(3):275–295.

Zohaib, M., and Nakanishi, H. 2018. Dynamic difficulty adjustment (dda) in computer games: A review. *Adv. in Hum.-Comp. Int.* 2018.