MyPAM Game Design

# Introduction

These two research papers give a vital introduction to the advantages of and design principles for video games in rehabilitation.

[The use and effect of video game design theory in the creation of game-based systems for upper limb stroke rehabilitation](https://journals.sagepub.com/doi/full/10.1177/2055668316643644)

[*Video Games and Rehabilitation: Using Design Principles to Enhance Engagement in Physical Therapy*](https://journals.lww.com/jnpt/Fulltext/2013/12000/Video_Games_and_Rehabilitation__Using_Design.4.aspx)

The point of making these computer games is to increase patient engagement with therapeutic exercises to maximise their recovery. Getting the patient to make the movements is the priority. Keep in mind that the point of making the games fun is to achieve this. This is most important when considering control schemes when designing each game.

The 6 key design principles for increasing player engagement are:

* Reward
* Optimal Challenge
* Feedback
* Choice/Interactivity
* Clear Instructions
* Socialisation

These are discussed in great detail in the second paper linked above.

Socialisation is the greatest challenge. There won’t be anything like the required player base for online multiplayer and the patients may not have internet. Local coop may be possible with a single device with clever game design. Competition would be difficult to balance due to the capability of patients varying greatly. Leaderboards would be dominated by the more capable, so wouldn’t provide much of a self-esteem boost to the less capable.

# Control Scheme

The movements made by the patient are translated into coordinates which are fed through a mid-level controller program to a script that the games can access. This script handles communication between the game and MyPAM. For instructions on using the script, see the Game Dev ReadMe.

The joystick assists the player with motors. To do this, the game must inform MyPAM of the target coordinates *before* the patient starts moving the joystick. The joystick can therefore not be used to make decisions.

In addition to the joystick, the patient will have a button. This can be used to make limited decisions. The patient can choose between pressing it, not pressing it, holding it down, and timing when they press it.

MyPAM currently requires a keyboard as well, which gives them access to the escape button which will exit the game. This feature is implemented in the UDP\_Handler script. Avoid using the keyboard for game functions.

The joystick movements need to produce relatable results in the game so that the patient can associate the game actions to moving their arm. MyPAM currently works in two dimensions, x and y. These can be mapped to produce a variety of outcomes. Examples include:

* Map linearly to a 2D plane (like in Monkey Bridge and Sky Striker)
* Map to rotations (like in the shooting game)
* Map to progression – one joystick position is 0% and another is 100%, moving between them moves the character by that amount (like in Maze)

Each of the games made in 2019 are 3D, but are controlled by x and y.

The device outputs the position of all three joints (one of which will stay constant). The third is most likely to be the main game input, but the second could form an interesting mechanic.

There are currently plans for four different trajectory controls for motor assistance. MyPAM determines the velocity and position of the motors based on the target coordinates from the game and the trajectory type.

**Point to point** takes a target position from the game and calculates a smooth velocity curve to make the motion comfortable for the patient.

**Curve/arbitrary** movement requires a number of adjacent positions from the game, which are moved to at a constant speed, hopefully resulting in a continuous motion in the shape of the curved path. The game sends the end position and the intermediate positions.

**Attractors and deflectors** affect the motor assistance. As with point to point, the game sends a target position as well as an array of attractors and deflectors, then the controller calculates the trajectory.

**Valley** is to be designed.

# General Game Features

To make sure games are all compatible and make development faster, it would be useful to reuse scripts and to use the same mechanisms for each game. The games should also have a similar external feel and control system, e.g. every game having a title screen that starts the game when the button is pushed. The UDP\_Handler is a good example of a reusable script that makes development easier – the game just needs to apply the script and have functions for mapping the joystick coordinates into game actions, and for sending target positions for assistance.

When designing games, consider:

* Title screen
* Level management
* Data saving
* General Settings (accessibility options, audio/visual, etc)
* Game specific settings (enable/disable hints, more specific audio control, etc)

Stroke patients tend to be older and may have other disabilities. These need to be considered when designing the game. [This YouTube series](https://www.youtube.com/playlist?list=PLc38fcMFcV_vvWOhMDriBlVocTZ8mKQzR) discusses this topic

# Summary

* Read the 2 papers
* Make the game engaging to encourage arm movements
* The joystick cannot be used to make decisions
* A single button can be used to make decisions
* The destination of the joystick needs to be determined before it is moved
* There are a few different options for trajectory towards this destination
* Get creative with how the joystick manipulates the game, but make the game action relate to the movement
* Try to make scripts that can be reused in multiple games – ideally each game would only take a week or 2 to implement things like artwork, game specific mechanics, and mapping the inputs to something tangible
* Design for disability