A model-based design tool for 3D GUI layout design that accommodates user attributes

Jamie Lee

Supervisors: Prof. Per Ola Kristensson
Dr. John Dudley

Department of Engineering
University of Cambridge

This dissertation is submitted for the degree of
Master of Philosophy in Machine Learning and Machine Intelligence

Churchill College

August 2021

Declaration

I, Jamie Lee of Churchill College, being a candidate for the MPhil in Machine Learning and Machine Intelligence, hereby declare that this report and the work described in it are my own work, unaided except as may be specified below, and that the report does not contain material that has already been used to any substantial extent for a comparable purpose.

This report contains 14,784 words, excluding declarations, bibliography, photographs and diagrams, but including tables, footnotes, figure captions and appendices.

The software used for this report was written entirely by the author except in the following cases:

1. The functions used to compute the consumed endurance, muscle activation, and RULA metrics as described in Section 4.1.1 were included in software written by Belo et. al. [15] , available at https://github.com/joaobelo92/xrgonomics.

2. All materials, assets, and prefabs used to construct the 'virtual' real environments as described in Section 4.1.2 were obtained for free from the Unity Asset Store.

3. The functions used to compute the colourfulness and edgeness metrics as described in Section 4.1.4 were written by Dudley et. al. [14]. Permission was gained from the author to use this software.

4. The Python package used as the basis for the Preference Learning application as described in Section 4.4 was written by Alkhassim et. al., available at https://github.com/chariff/GPro.

Repositories for the original software written for this project are provided in Appendix C.

Jamie Lee
August 2021

Acknowledgements

First and foremost I would like to thank Prof. Per Ola Kristensson and Dr. John Dudley. They have offered me continued support and guidance throughout the past several months of my dissertation. From their mentorship, I have gained an appreciation for the HCI community and expanded my knowledge of designing intelligent interactive systems.

I would also like to thank Michael Morse, Executive Director of the The Winston Churchill Foundation of the United States. I am beyond grateful for having the opportunity to study at the University of Cambridge under the Churchill Scholarship. Through this experience, I have met countless individuals from around the world with pursuits in various fields of research. The immersion in such an intelligent, forward-thinking, and imaginative community has truly opened my eyes to the value of higher education.

My friends at Churchill College have been the primary reason for my memorable experience while at Cambridge, and I thank them for their kindness and support. I will always remember the laughs and memories shared during our extravagant cooking escapades, late-night conversations, and adventures in downtown Cambridge.

Finally, I am thankful for my family for their never-ending love and words of encouragement. Although there have been many arduous points during this academic year, they have cheered me on during the most challenging of times. My mother, father, and younger brother have all been influential sources of motivation throughout my academic journey at Cambridge.

Abstract

The designing of inclusive and immersive user interfaces for virtual and augmented reality (VR, AR) systems remains a challenge for the Human-Computer Interaction community. Design exclusion, coupled with limited research in understanding the usability of VR/AR systems for individuals of various (dis)abilities, is detrimental to the goal of providing an enriching and inclusive experience for all users. Furthermore, the designing of optimal 3D user interfaces (UIs) is prone to noisy behaviour from users and variability between user preferences, which makes this process non-trivial.

This dissertation provides three central contributions to address this challenge. First, we systematically investigate the design parameters that dominate user performance and comfort when interacting with UI layouts using techniques borrowed from design engineering. Second, we create a novel model-based design toolkit that facilitates the design, creation, and exploration of inclusively immersive 3D UI layouts. This toolkit can be used to construct UI layouts that accommodate the unique perceptual, cognitive, and physical capabilities of the user. We successfully use the design parameter analyses to convert the parameters into predictive models, which are then used to construct a single objective function to optimise. Additionally, we apply a user-in-the-loop approach to our toolkit through preference learning to integrate the designer's feedback into the optimisation process and suggest alternative configurations pertaining to user capability at design time. Third, we demonstrate a method of evaluating the usability of the toolkit by constructing several design tasks and specifying criteria relating to the Cognitive Dimensions of Notations. The author and another expert evaluator complete these tasks to evaluate the toolkit and provide qualitative feedback and potential extensions to the project.

Our UI design toolkit differs from current state-of-the-art techniques for several different reasons: while previous works have integrated specific types of human performance models (such as those related to visual aesthetics or physical ergonomics) into toolkits, our toolkit is the first to combine models related to physical ergonomics, cognition, and visual perception into a single product. Furthermore, the toolkit utilises both multi-objective optimisation and preference learning to provide the designer with the ability to directly influence the optimisation process and steer the direction between exploration and exploitation. It is hoped

that the identification of critical design parameters and the preliminary framework provided by the design toolkit can provide a foundation for further work to improve the quality of immersive experiences delivered to users of various abilities.

Table of contents

List of figures

List of tables

Nomenclature

Acronyms / Abbreviations

AR	Augmented Reality
CD	Cognitive Dimensions
CE	Consumed Endurance
GP	Gaussian Process
GUI	Graphical User Interface
HCI	Human-Computer Interaction
HMD	Head-Mounted Display
ID	Index of Difficulty
LOD	Level of Detail
MOO	Multi-objective Optimisation
MT	Movement Time
RULA	Rapid Upper Limb Assessment
UI	User Interface
V&V	Verification & Validation
VR	Virtual Reality

Chapter 1

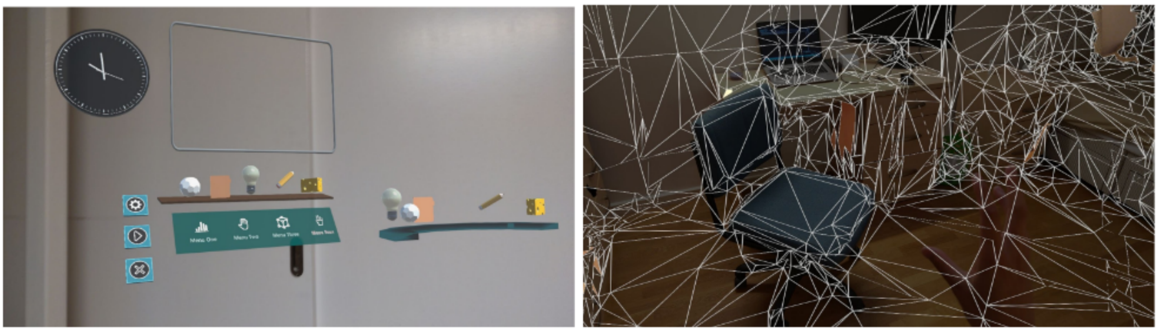
Introduction

1.1 Motivations and Contributions

The past several decades have welcomed improvements in virtual and augmented reality (VR, AR) systems, providing users with an immersive and interactive digital experience. While these systems have been integrated into devices such as mobile devices, projectors, and PCs, the integration into head-mounted displays (HMDs) has facilitated a wider range in movement and a more comfortable user experience. These devices can provide a completely immersive simulated experience as with VR, or a projection of computer-generated objects onto the real world as with AR. The multiple sensory immersion offered by both devices enables users to bridge the divide between the physical and virtual worlds and have experiences much different from reality. For example, the Hololens 2 is an optical see-through HMD device shown in Figure 1.1 which provides an AR experience by projecting holograms onto the user's real world environment. Figure 1.2 (a) displays an image taken with the camera attached to the Hololens 2 of a sample UI consisting of various widgets and buttons which the user can physically interact with. The Hololens 2 also utilises spatial mapping, which provides a detailed representation of real-world surfaces in the environment around the device as shown in Figure 1.2 (b).

Due to the importance of providing a comfortable and efficient VR/AR experience, there has been increased focus in the study of physical and visual ergonomics for designing VR/AR games and applications. For Human-Computer Interaction (HCI) research, there are two major challenges associated with these systems. First, designers cannot be aware of the capabilities of every user for the application they are designing. This limitation is conducive to design exclusion, in which the designer unintentionally excludes certain users or user groups due to implicit biases or assumptions they form about the user. Design exclusion, coupled with limited research in understanding the usability of VR/AR systems

Fig. 1.1 The Hololens 2 optical see-through HMD.



(a) (b)

Fig. 1.2 (a) A sample UI designed for the Hololens 2 using Unity, (b) the spatial mapping feature of the device.

for individuals of various (dis)abilities, is detrimental to the goal of providing an enriching and inclusive experience for all users.

A second challenge for VR and AR systems is the designing of effective 3D user interface (UI) layouts. The 'optimality' of these layouts is often variable to human perception, psychology, and preference; an optimal design for one user will usually not be optimal for all other users. Such noisy behaviour from users and variability between user preferences makes the process of UI optimisation non-trivial.

To mitigate designer bias and facilitate the design, creation, and exploration of inclusively immersive user interfaces (UIs), this project aims to create a model-based design toolkit for building 3D UI layouts which considers the perceptual, cognitive, and physical capabilities of the user. Because the 3D UIs designed with this toolkit may be used by individuals with some form of impairment, a central focus of this project will be the identification and analysis of controllable and uncontrollable design parameters which dominate user performance and comfort when interacting with the interface. Specifically, the key objectives are:

1. Systematically investigate the controllable and uncontrollable design parameters that dominate user performance and comfort when interacting with 3D UI layouts via a model-based approach.

2. Develop a model-based design tool for constructing 3D UI layouts that can provide feedback and suggest alternative configurations pertaining to user capability to the designer at design time.

Our UI design toolkit differs from current state-of-the-art techniques for several reasons. First, it takes a model-based approach to parameterise the various perceptual, cognitive, and physical factors which affect user performance and comfort when interacting with 3D UI layouts. The UI toolkit allows the designer to continuously choose and adjust the weights of these functions, which are then used in multi-objected weighted optimisation to search for an optimal UI layout. This first feature enables the incorporation of a variety of human performance models ranging from physical ergonomics to visual perception, which has not been yet accomplished by previous UI toolkits in HCI research.

Second, the toolkit leverages a user-in-the-loop approach through preference learning to integrate the designer's feedback into the optimisation process. The user-in-the-loop approach incorporates user participation into the feedback loop, and the coupling between this approach and preference learning enables a balance between exploration of new designs while exploiting preferences and previous design knowledge. From this second feature, the toolkit provides the designer with the ability to directly influence the optimisation process and steer the direction between exploration and exploitation, which is another novel aspect of our toolkit.

1.2 Thesis Outline

We structure the remainder of this thesis as follows. In Chapter 2, we provide the necessary background for the rest of the thesis. We first introduce VR and AR and potential challenges that arise when designing for these devices. Next, we explore the foundations of ability-based design and design engineering. Finally, we describe the concepts of Bayesian optimisation, user-in-the-loop design, and preference learning.

In Chapter 3, we describe our parameter analysis approach to the project. We analyse each of the design parameters which dominate user performance and comfort when interacting with 3D UI layouts on VR/AR devices. For each, we describe how the parameter may be modelled or further investigated through envelope analysis to simulate the system's potential performance under a range of parameter choices.

Next, we describe the method used to design and implement our model-based UI design toolkit in Chapter 4. Specifically, the conversion of each parameter into a function model and the formulation of an objective function is explained in this section. Furthermore, we describe our integration of multi-objective weighted sum optimisation, Bayesian optimisation, and preference learning into the process of generating an optimal UI layout for AR devices.

In Chapter 5, we provide a discussion of our evaluation goals, method, and criteria used to verify and validate our UI design toolkit. In the following chapter, we discuss the toolkit's design implications, limitations, and potential extensions. Finally, we conclude with closing remarks and aims for future work.

Chapter 2

Background

2.1 Virtual and Augmented Reality

Improvements in device affordability and quality of VR/AR applications, coupled with growing interest in VR/AR development has enabled more users to familiarise themselves with this technology. Despite these advances, the creation of inclusive VR/AR applications and interfaces remains a challenge. As with many devices such as computers and mobile phones, these systems are often designed with certain ability assumptions. For example, consider the process of putting on a VR headset: one may need to insert batteries, plug cords into a computer, or adjust the straps on the headset. Although the task may seem simple to some users, others may find it challenging. A study conducted by Mott et. al. [41] in the form of semi-structured interviews with individuals with limited mobility revealed that the abilities of many of their participants did not match the assumptions embedded in the current VR design. The researchers found that many participants struggled with one or multiple of seven VR accessibility barriers, including manipulating dual motion controllers, putting on and taking off the VR HMDs, and setting up the VR system. These barriers often deter users from engaging with such devices and emphasise the need for designing VR/AR systems which are accessible to all people. For this project, we will focus primarily on AR technology and the goal of improving the accessibility of these systems for people of all abilities.

2.2 Ability-based Design

A common goal shared by many developers is to create technology accessible to as many users as possible. Despite ongoing research in accessible computing, many UIs fail to provide similar experiences for individuals with disabilities and/or centralise on the notion of disability, rather than ability. However, the abilities of a user can span across a vast spectrum and even fluctuate given the circumstance or environment they are in. Furthermore, anyone may face a reduction in abilities, even those who may not normally be considered disabled. For example, alcohol impairs the ability to drive and operate equipment, but only during the period of time in which the individual is in a state of drunkenness. An even simpler scenario may be when a person holds several books in their arms, temporarily limiting the use of that arm. Thus, assumptions made about a user's (dis)abilities may make it more challenging to design more inclusive applications.

An alternative approach to inclusive design is to refocus on ability-based design [56]. Instead of considering what a person cannot do, ability-based design asks the question, "what can a person do?" This concept differs from universal design, which aims to develop systems for general use with a 'one size fits all' mentality. Instead, systems developed through ability-based design may try to adapt and tailor themselves to the needs and preferences of a specific user or user group. For example, SUPPLE [19] is an ability-based system which generates different renditions in response to different user usage patterns. The system automatically constructs UIs using an optimisation process which searches the design space for an interface which minimises the users' movement time. The model for movement time is created by prompting the user to complete a series of clicking, pointing, dragging, and list selection tasks. Through this approach, SUPPLE generates UIs customised to a users' abilities which enables more efficient and accessible mouse interactions.

In order to make VR/AR systems more accessible to users with varying degrees of perceptual, cognitive, and physical capabilities, we must understand the factors which affect user comfort and performance when interacting with these systems. There have been many studies involving human participants with various mobility impairments using VR/AR systems. Mott et. al. [41] conducted a semi-structured interview study with participants with mobility limitations affecting head, arms, hands, and/or legs. The participants were asked questions regarding prior experiences with VR systems to better understand the challenges they might encounter, and although not all participants expressed the same concerns, their concerns reflected the need to consider the abilities of users with limited mobility in the design process for VR applications.

Interviews and questionnaires are common methodologies for determining such parameters within the HCI research community. Specifically, semi-structured quantitative studies are often employed in the understanding of user needs and behaviours when using interactive technologies. Blandford [5] address the principles for designing, conducting, and reporting on such qualitative studies for the purpose of understanding current needs and practices and evaluating the effects of new technologies in practice. We see these principles reflected in many fields across HCI; for example, Dias et. al. [13] interviews patients with Parkinson's Disease (PD), physicians, and software/game developers to identify the most significant game-design factors in designing assistive HCI serious games for PD patients.

Although controlled user studies are useful for the evaluation stage, it is often infeasible for designers to effectively explore a large parameter space with individuals representing all forms of capabilities. Furthermore, user experiences and behaviour may be inconsistent and vary over time. Another possibility is to generate realistic data from proxy-users (users who can impersonate other users or mimic their abilities). However, the user experience for those with some form of impairment may not be accurately reflected by using proxy-users. Thus, alternative methods to controlled user studies may be necessary in cases involving larger parameter spaces.

2.3 Design Engineering

Given the difficulty in extracting data in-situ from actual users or generating realistic data from proxy users, we instead integrate techniques adapted from design engineering, a methodology used in engineering to design products and systems which is often useful for systems which are complex and costly to validate. Having identified a function model, it is possible to parameterise this model. Specifically, we conduct parameter analysis to identify and analyse the controllable and uncontrollable parameters which dominate user performance and comfort when interacting with UIs. While both forms of parameters govern function execution, uncontrollable parameters cannot be directly influenced and set by the designer as with controllable parameters. Investigation of these parameters enables simulation of a system's potential performance through investigation of a range of parameter choices in a process known as envelope analysis. Kristensson et. al. [29] conducts envelope analysis and studies theoretical performance envelopes of a context-aware sentence retrieval system. By extracting parameters from the functional description of the system and simulating its potential performance, they are able to identify potential keystroke savings as a function of the parameters of the subsystems, revealing additional insight in designing for augmentative and alternative communication technologies.

After parameterisation of a model, designers often aim to find the most optimal settings of their controllable parameters to maximise efficiency in terms of their design objectives. However, layout optimisation is a complex task, especially when the task encompasses both usability as well as aesthetic qualities. Recent research has employed a model-based UI optimisation approach to optimise UIs and improve designs towards specific objectives. Unlike heuristic methods, this approach uses design knowledge in the form of user simulations, models, and/or heuristics as an objective function to model how users interact with and perceive such layouts. Todi et. al. [52] adapts this method to develop Sketchplore, an interactive layout sketching tool with a real-time layout optimiser to generate usable and aesthetic layouts for designers. Their design tool uses predictive models to address the aesthetic and sensorimotor performance measures of generated layouts, such as visual clutter and search, grid quality, colour harmony, and target acquisition, to define a multi-objective function. Multi-threaded optimisation is then used to explore and exploit the design space.

2.4 Bayesian Optimisation

While multi-objective optimisation techniques such as that used in Sketchplore enables the tool to find the most optimal setting of various parameters, designers may not want to be limited to a single optimal design. Instead, it may be more useful to explore multiple optimal options across the parameter space. Bayesian optimisation is a machine learning technique which is well suited for this purpose. Specifically, this technique enables exploration of cost functions which are expensive or difficult to evaluate; thus, it is useful in supporting UI design, since this process usually involves multiple objectives which would be optimised by user evaluation. Detailed formulation of the basic principles of Bayesian optimisation is given by Snoek et. al. [46]. This technique has been commonly employed in HCI research; for example, Brochu et. al. [6] integrates Bayesian optimisation to generate various virtual smoke animations, which are then displayed in a preference gallery. Furthermore, Dudley et. al. [14] leverages a combination of Bayesian optimisation and crowdsourcing to refine design parameters for a range of UI designs. Shahriari et. al. [45] provides several other practical applications of this technique such as A/B testing, robots and reinforcement learning, natural language processing, and so forth.

In Bayesian optimisation, we are interested in attaining the minimum of an unknown function $f(\mathbf{x})$, known as the objective function. This function is commonly modelled as a Gaussian Process (GP), which describes a distribution over functions with $f(\mathbf{x})$ in terms of its mean function $m(\mathbf{x})$ and covariance function $k(\mathbf{x},\mathbf{x}')$ [43]:

$$m(\mathbf{x}) : \mathbb{E}[f(\mathbf{x})] \tag{2.1}$$

$$k(\mathbf{x},\mathbf{x}') : \mathbb{E}[f(\mathbf{x}) - m(\mathbf{x}))(f(\mathbf{x}') - m(\mathbf{x}'))] \tag{2.2}$$

We may write the Gaussian Process as:

$$f(\mathbf{x}) \sim GP\left(m(\mathbf{x}), k\left(\mathbf{x}, \mathbf{x}'\right)\right) \tag{2.3}$$

We assume that function $f(\mathbf{x})$ is drawn from a GP prior; the mean function $m(\mathbf{x})$ and covariance function $k(\mathbf{x}, \mathbf{x}')$ are essentially the function equivalents of the mean and variance of a random variable (in this case, the function $f(\mathbf{x})$). $k(\mathbf{x}, \mathbf{x}')$ may also be referred to as the kernel, which specifies the covariance function of the GP. The kernel consists of hyperparameters, which are optimised during the process of fitting observation data to the GP. The observation data, in the case of UI optimisation, represent observations of user performance when interacting with the UI. There are various kernels to choose from, each with different properties to reflect different assumptions about the underlying data.

After fitting the GP to the observation data, the next step in Bayesian optimisation is to use the GP to probabilistically determine the next point to evaluate. This step involves an acquisition function, which reflects the desirability or goodness of evaluating the function at a certain point. Some examples of acquisition functions include probability of improvement, expected improvement, upper confidence bounds, Thompson sampling, and combinations of the previous methods. The determination of the next point to sample influences the balance between exploration and exploitation. While it may be desirable to explore regions in which we have limited information, especially in early stages of optimisation, it is often beneficial to utilise information learned during the process once more points have been observed. Furthermore, it is also useful to exploit regions in which we are more knowledgeable about; however, this may come at the expense of missing regions which may yield better outcomes. Thus, the balance between exploration and exploitation is often difficult to perfect.

2.5 User-in-the-Loop Design

From a Bayesian optimisation framework, we may consider the UI optimisation process as a balance of exploration of new designs while exploiting preferences and previous design knowledge. The key challenge with designing an optimal UI is that the ‘optimality’ of such designs are variable to human perception, psychology, and preference. Such noisy behaviour from users and variability between user preferences makes this process non-trivial. Even after obtaining an optimal solution through Bayesian optimisation, it cannot be guaranteed that the solution will be the design that is desired for all users in all scenarios. Fortunately, the integration of user feedback into the optimisation process can help facilitate this balance of exploration and exploitation: this process is known as user-in-the-loop design.

The amount of user involvement is an important consideration which may affect the generation of an optimal solution. For example, allowing users to operate with complete autonomy may be detrimental in certain cases, especially when they have mistrust in the application or conceal their actions [16]. Furthermore, possessing more control may stimulate more anxiety or discomfort for some users, as expressed by van der Heijden [55]. Furthermore, the timing of user input requests should also be considered in the user-in-the-loop process. Losing et. al. [31] employs this process in the labelling of obstacles encountered by a robot, but describes the challenges associated with the timing of captured training instances. A robot drives randomly and encounters various objects, and the user interacts in real-time with the robot by labelling approached objects with an iPad. Because the user may choose which objects to label and incrementally incorporate into the model, there is potential of violating independent and identically distributed (i.i.d.) assumptions in the model.

With careful consideration of the extent and timing of user feedback in the feedback loop, user-in-the-loop approaches possess potential for developing applications and systems which enhance user capabilities. This process has been previously integrated in HCI applications; for example, AppGrouper [7] incorporates human input in a knowledge-graph-based clustering process to allow domain experts to steer the clustering process in early, mid, and late stages. In evaluation, the quality of clustering results were shown to improve when enabling users to directly edit clusters in comparison to generating clusters by algorithm only.

2.6 Preference Learning

One potential method of integrating designer feedback into the optimisation process is through a method known as preference learning, which utilises the preferences of the designer in the process of generating an optimal solution. Specifically, we take a probabilistic kernel approach to preference learning based on Gaussian processes, as proposed by Chu and Ghahramani [9] in 2006. The overall goal is to learn the underlying ordering over pairwise preferences between instances (the training data). We may consider a set of n distinct instances $x_i \in \mathcal{R}^d$ denoted as $\mathcal{X} = \{x_i : i = 1, ..., n\}$, and a set of m observed pairwise preference relations on the instances, denoted as:

$$\mathcal{D} = \{v_k \succ u_k : k = 1, \dots, m\} \tag{2.4}$$

where $v_k \in \mathcal{X}, u_k \in \mathcal{X}$, and $v_k \succ u_k$ means the instance v_k is preferred to u_k. In the application of preference learning to UI design, v_k and u_k represent two UI designs with different arrangements of widgets.

The central idea is to assume that there is an unobservable latent function value $f(x_i)$ associated with each training sample x_i and that the function values $f(x_i)$ preserve the preference relations observed in the dataset. These latent function values are assumed to be a realisation of random variables in a zero-mean Gaussian process.

The method of capturing the designer's preference is summarised as follows:

1. Query the designer with a paired comparison between two UI designs and record the choice.
2. Update the Gaussian process model with the choice made by the designer.
3. Optimise a utility function which seeks a balance between exploration and exploitation of the latent function.

Through this process, we may capture preference relations in a Bayesian framework, allowing for global optimisation of the latent function values $f(x_i)$ describing each preference relation x_i.

Chapter 3

Parameter Analysis

This portion of the project identifies the relevant controllable and uncontrollable parameters which dominate user performance and comfort when interacting with 3D user interfaces. Due to the difficulty in extracting data in-situ from actual users or generating realistic data from proxy users, we take a *model-based* approach. This approach involves two steps: (1) identification and examination of pertinent models of human performance, and (2) determination of the optimal settings of controllable parameters using these models. A model-based approach offers potential for cost and time-effective evaluation of user performance without the need for intrusive measures. This approach has been used previously in UI development; for example, SPRWeb [18] is a tool that recolours websites to preserve subjective responses and improves colour differentiability to enable users with colour vision deficiency (CVD) to have similar online experiences as non-CVD users. Flatla et. al. [18] use models of subjective responses from external studies and develop a constraint optimisation technique which seeks to minimise a cost function computed by a weighted sum of four individual costs: perceptual naturalness, perceptual differentiability, subjective-response naturalness, and subjective-response differentiability. Their evaluation demonstrated that SPRWeb outperformed the state-of-the-art Kuhn recolourer in choosing replacement colours for recolouring websites. Sketchplorer [52] is another example of a model-based approach; the sketching tool uses a real-time layout optimiser which uses predictive models of sensorimotor performance and perception to steer the designer toward more usable and aesthetic layout designs.

In the following subsections, we will identify and describe relevant uncontrollable and controllable parameters which affect user performance and comfort when interacting with UIs designed for AR systems. After identification of optimal parameter settings, we will integrate these settings into our UI design toolkit and enable the toolkit to generate the most optimal UI given specifications set by the designer.

3.1 Uncontrollable Parameters

We first identify and describe relevant parameters which cannot be directly set by the designer, also known as *uncontrollable parameters*. These parameters are relevant in the UI optimisation process and can be used to construct and mitigate the occurrence of a potential “worst case" scenario. We adapt quantitative models of these uncontrollable parameters from various studies to analyse the effect of each parameter on user performance and comfort when interacting with UIs designed for AR systems.

3.1.1 Physical Ergonomics

User comfort and ergonomics is an important consideration for HCI for the purpose of improving the user experience when interacting with VR/AR systems. Despite ongoing research, there are still challenges with evaluating VR/AR ergonomics; current methods often involve interviews and/or questionnaires such as those used by Mott et. al. [41] to evaluate the accessibility of VR systems for persons of limited mobility. In most scenarios, designers will not be able to obtain feedback, if any, from enough individuals to adequately represent all potential target users. Thus, researchers have attempted to find methods of quantitatively modelling and predicting user ergonomics. In the following subsections, we describe three methods of quantitatively analysing physical ergonomics: consumed endurance, biomechanical simulation, and Rapid Upper Limb Assessment (RULA).

Consumed Endurance

VR/AR devices commonly use arm and hand gestures to enable communication between the user and system. However, prolonged use of the arms and upper body for mid-air gestures often leads to upper arm fatigue, a phenomenon commonly known as the ‘gorilla-arm effect’. Hincapié-Ramos et. al. [25] develops a metric to quantify the severity of this effect, Consumed Endurance (CE), which is derived from the biomechanical structure of the upper arm. Although multiple body parts are involved in such mid-air arm interactions, Hincapié-Ramos et. al. focus on the shoulder joint since it largely dominates the forces required for moving the arm. Therefore, this perspective of CE considers endurance of the shoulder in terms of torque as a ratio to the interaction time and uses shoulder torque as an index for muscle strain. To further simplify CE computations, we assume that all arm poses are static, since the shoulder must match the gravity torque when the arm is static and the arm’s torque and angular acceleration are equal to zero.

Biomechanical Simulation

The prediction of posture, location, direction, degree, and other factors of human movement often involves intrusive and/or tedious procedures. Fortunately, biomechanical simulation has offered a means of capturing this information and enables cost-efficient estimation of physical ergonomics. It has potential for indicating user fatigue and ergonomics in a non-intrusive manner, which is useful for HCI applications and VR/AR technology. The collection of optical motion tracking data for biomechanical simulation usually involves a mapping of physical to virtual markers, scaling of the musculoskeletal model, adjustment of markers through inverse kinematics, and estimation of the muscle activations [3]. We adopt the method implemented by Belo et. al. [15] in the estimation of muscle activations from biomechanical simulations. This method uses simulations from OpenSim 4.1 [12], an open-source tool for biomechanical modelling and simulation, as well as the upper extremity model created by Saul et. al. [44]. Belo et. al. [15] analyse each arm pose over time and then save the timeframe which minimises the reserve actuation for each pose, which yields an activation value for each muscle and reserve actuator in the model. These values are then combined into a single cost function to describe the cost of each arm pose in terms of muscle activation.

Rapid Upper Limb Assessment

Rapid Upper Limb Assessment (RULA) is a heuristic survey method developed by McAtamney et. al. [38] to provide a quick assessment of the postures of the neck, trunk, and upper limbs, along with muscle function and external loads experienced by the body. To allow for easy identification of posture ranges, the range of movement for each body part is divided into sections, which are then numbered; low posture scores reflect postures with minimal risk factors, while higher scores represent more extreme postures and an increased presence of risk factors. We use scores from the study for the upper arm, lower arm, and wrist, which are based on the joint angles of the upper and lower arms.

3.1.2 Cognitive Load

A key feature of AR devices is the ability to project computer-generated visuals onto the users' real environment. Because the user is still situated in their current environment, there is more consideration of the contextual details associated with this environment in comparison to a fully immersive experience offered by VR technology. The users' context may include environmental conditions (e.g. indoors vs outdoors), task, and cognitive load. For example, experimenting in a laboratory with equipment and other researchers would demand a higher cognitive load than sitting alone in an office. In the first environment, the user may desire an interface with fewer visual details in comparison to the second.

Cognitive Load Theory [50], which involves estimation of the users' workload, is an important aspect of HCI and the development of interactive systems. Generally, designers would want to limit distractions and overloading users with information. However, it is often difficult for designers to be aware of the cognitive abilities of each user and furthermore develop interfaces which can adapt to changing cognitive levels. Current research has explored various methods of inferring the users' cognitive load in relation to HCI applications: these methods generally fall into one of three predominant categories, the first of which involve subjective measures such as the NASA TLX [22], a commonly-used questionnaire which assesses subjective mental workload on a multi-dimensional rating scale. These subjective measures may be time-consuming and tedious however, and users may forget various details of the tasks they are questioned about. The second category for measuring cognitive load includes physiological measures such as heart rate variability, electromyography, and skin conductance. However, a key challenge with such measures is that they are invasive and rely on physical contact with the user. The final category, eye tracking, offers the best potential for non-invasive estimation of cognitive load. Gaze tracking and pupil dilation have been previously researched and suggested to be related to the mental difficulty of tasks. This idea is often traced back to the study by Hess and Polt [24] demonstrating correlation between pupil size and mental activity in the form of simple multiplication problems. Lindlbauer et. al. [30] adopts this method of computing the frequency of changes in pupil diameter to estimate the cognitive load of the user when interacting with a UI generated with an HTC Vive Pro VR headset. This estimation of cognitive load is used to optimise the UI in terms of the amount of information provided (the level of detail, or LOD). While improvements in accuracy and lowered costs of eye trackers have increased their popularity, eye tracking methods may still suffer from practical limitations and errors caused by ambient light [4] and off-axis distortion [36].

3.2 Controllable Parameters

We now identify and describe relevant parameters which can be directly set by the designer, also known as *controllable parameters*. These parameters enable optimisation towards design objectives, or specifically, the construction of a UI layout which is adapted to the physical, perceptual, and cognitive abilities of the user.

3.2.1 Target Acquisition

The modelling of human movement is a major component of predicting human-computer interaction and ergonomics. Fitts' Law, which enables predictive modelling of human movement, is arguably the most commonly used human performance model in HCI. In his 1954 paper, Paul Morris Fitts [17] proposed a metric to quantify the difficulty of a target selection task. The metric was based on information theory, in which the difficulty of a task can be measured using the information metric bits, and that in carrying out a movement task information is transmitted through a human channel [32]. The measure of Fitts' index of difficulty (ID), in bits, is:

$$ID = \log_2(\frac{2A}{W}) \tag{3.1}$$

where the distance to the centre of the target (A) is analogous to a signal and the tolerance or width of the target (W) is like noise. The time to move to (MT) and select (e.g. hit, click, or press) a target of width W at amplitude A is:

$$MT = a + b \times ID \tag{3.2}$$

where a and b are constants which are determined empirically. Numerous other variants of the original Fitts' formulation have emerged based on his extensions, including the Shannon's formulation introduced by Scott MacKenzie in 1992 [32]:

$$MT = a + b \times \log_2(\frac{A}{W} + 1) \tag{3.3}$$

The adding of 1 instead of multiplying with 2 in the Shannon's formulation was introduced to guarantee positive values for the ID, as written in MacKenzie's published theory [33]. This formula has become popular in HCI and is often the variant of Fitts' Law used in research.

In terms of user interface design, Fitts' Law has often been employed to describe the time taken to click or interact with a widget using a mouse cursor or other device. Although it has been commonly applied to 1D and 2D tasks, the necessity for a 3D performance metric has facilitated recent extensions to 3D applications. However, the extension to 3D space introduces many complications, the first of which may be attributed to impaired depth perception. In VR/AR systems, the estimation of distances between the user and virtual targets usually differs from that of the user and physical target, which may cause them to overestimate their depth perception. Secondly, there are translational and rotational complexities for modelling target acquisition for 3D targets, since these targets can be arranged and manipulated across three axes as opposed to one or two [10]. Finally, interactions with targets in 3D UIs include more than clicking; users may point, grab, or even gaze at targets to form an interaction.

3.2.2 Colour Harmony

The use of colour in UIs spans across many purposes, some of which include to draw attention to certain elements, label or group items, or visualise similarity or differences between elements. This topic has also been explored in HCI with regards to human ability; for example, Chroma [51] is a wearable AR system based on Google Glass which automatically adapts the scene based on the type of colour blindness and allows users to see a filtered image of the current scene in real-time.

Colour also has the ability to impact human perception, and certain colours may invoke moods and feelings from the viewer. From an HCI perspective, the colouration of a UI layout may impact the degree to which a user finds the layout aesthetically pleasing or displeasing. When the placement of two or more colours generates a pleasant response, the colours are said to be in harmony. The exact definition of colour harmony is not clearly delineated, however. For centuries, artists have studied the balance and positioning of colours which evokes a sense of harmony. These methods have often lacked robust scientific methodology and have been subject to the discretion of the artist, thus creating many different definitions of the concept of colour harmony.

The introduction of quantitative representations of colour in the 20th century has helped shape the modern concept of colour harmony. In 1931, the CIE 1931 RGB colour space and CIE 1931 XYZ colour space were created by the International Commission on Illumination to become the first quantitative method of colour specification. However, the CIE colour spaces were still limited by the fact that distances in CIE space do not correspond equally to perceptual steps in colour. In 1943, Moon and Spencer [39] created a metric colourspace through mathematical transformation of the CIE space, leading to the development of a scientific theory of colour harmony. They later introduced a geometric formulation of classical colour harmony based on the principle that any arrangement of colours that can be sensed as an orderly combination will be pleasing. The general method followed was to reduce the problem to one of geometry in ω-space, or specifically, a colour space in which curves of constant hue appear as straight lines and the curves of constant chroma are uniformly spaced circles. In ω-space, various types of harmony correspond to geometric figures in space such as straight lines, triangles, and circles [40].

The definition of colour harmony was further augmented by Itten [27], who introduced a new colour wheel with emphasis on hue in 1960. This hue wheel of twelve colours was based on Itten's colour harmony theory, which defined complementary pairs as harmonious for all three-colour combinations whose colours form equilateral or isosceles triangles and all four-colour combinations forming squares or rectangles [27]. Based on these harmony schemes, Matsuda [37] introduced a set of 80 colour schemes in 1995 by combining several types of hue and tone distributions, to which our notion of colour harmony is based. These colour schemes have been integrated into various harmonic templates as described by Cohen-Or et. al. [11]. Each of these harmonic templates specify a range of colours within 'wedges' defined on an HSV colour wheel as shown in Figure 3.1. These colours may consist of shades of the same colour, or shades which are complementary to one another.

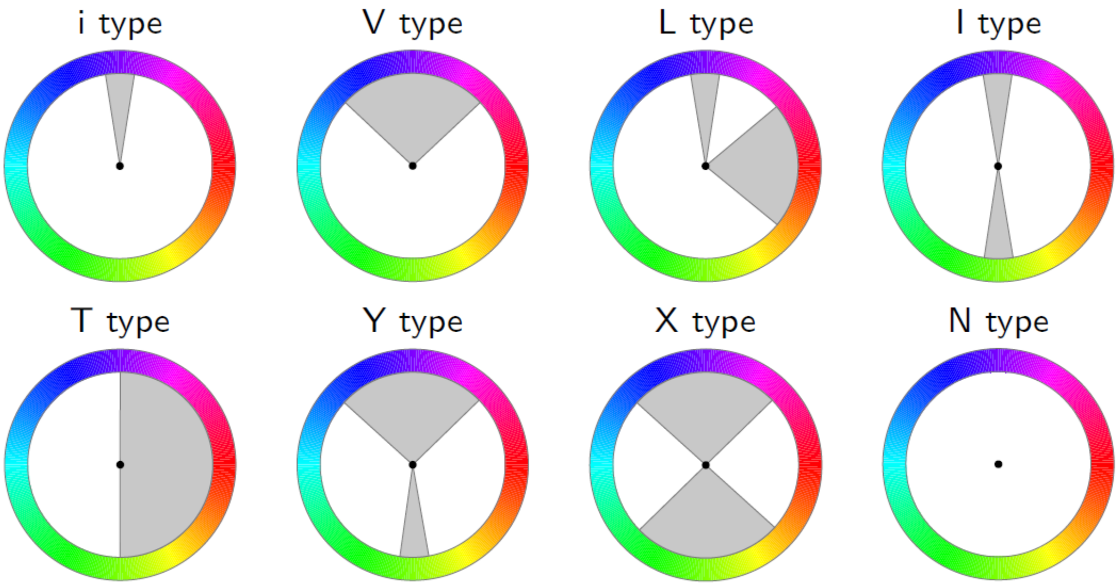


Fig. 3.1 Harmonic templates on the HSV hue wheel as given by Cohen-Or [11]. Colours which fall in the grey areas are considered to be harmonic.

3.2.3 Text Legibility

UIs designed for AR technology will usually consist of some form of textual content. Although the ability to overlay this virtual content onto real objects in the environment is a hallmark of AR, it may also induce challenges with the placement and design of text for these systems. For instance, text legibility may be an issue due to the interaction between the content and the texture in the background. The switching of the users' focus between the real environment and overlaid virtual data is known as competitive see-through [1] and is correlated to the users' comfort and usability of the application. Limited text legibility can spoil the AR experience and its effectiveness in conveying content for the user; thus, methods of designing and placing virtual text content have been widely studied in the HCI field. For example, Manghisi et. al. [34] outlines three distinct strategies for improving text legibility for AR systems: 1) adjust the text placement, 2) adjust the text appearance, and 3) place a panel behind the text.

We base our strategy for improving the text legibility of UIs created with our toolkit on the idea that highly colourful and/or noisy regions in the users' environment should be avoided when placing UI components in a layout. Thus, we incorporate concepts taken from the work of Dudley et. al. [14], which investigates this idea in the development of contextually-adaptive text content for AR. Dudley et. al. [14] utilises crowdsourcing to capture user preference data with regards to placement and colouration of text panels in maximising text legibility. The label placement locations collected from the users are then scored based on texture colouration under the hypothesis that a highly colourful background region will be avoided when placing the label. Dudley et. al. [14] adopts a simple colourfulness metric M introduced by Hasler and Suesstrunk [23], as well as an edgeness metric F is computed to quantify the degree of texturing or 'busyness' of an image [49]. For our toolkit, we consider both M and F of regions in the users' environment to determine the best placement of content in the UI.

Chapter 4

UI Toolkit Design

We design our UI design toolkit for the Hololens 2, an optical see-through HMD device shown in Figure 1.1. The Hololens 2 provides an AR experience by projecting holograms onto the user's real world environment. We define the interaction space of this AR system as a 3D Cartesian grid consisting of positions a human can reach and manipulate objects with a fixed torso position, analogous to the method used by Belo et. al. [15]. The interaction space is discretised into elements called *voxels*. The voxel dimensions are initialised with a default side length of 10cm, but can be adjusted to change the granularity of the interaction space representation. These voxels are used to determine the optimal placement of UI elements in terms of physical ergonomics, text legibility, and colour harmony, as described in the next subsections.

To simplify real-time computations and reduce overhead, the UI toolkit operates under three assumptions: (1) the only interaction the user makes with the UI are pressing widgets with their finger (a touch press), (2) the users' environment is static, and (3) the user is in a static position. We discuss the implications of these assumptions in detail in Section 6.2.

Repositories for the original software written for this project are provided in Appendix C.

4.1 Predictive Models

Using our analyses of the effects of each controllable and uncontrollable parameters on human performance, we may convert each parameter into a predictive model. We describe the methods used to quantify the cost of such effects in the subsections below.

4.1.1 Physical Ergonomics

We adopt methods used by Tolani et. al. [53] to simplify the inverse kinematics procedure for a seven degree of freedom model of the human arm. The human arm is a complex mechanical structure which is difficult to model accurately. However, the arm can be modelled as a two-segment chain, in which the forearm and wrist constitute a single segment. Considering the scenario in which the wrist (the end-effector) and shoulder positions are fixed, the elbow is free to swivel about an axis from the wrist to the shoulder. As the swivel angle ϕ varies, the elbow traces the arc of a circle which lies on a plane whose normal is parallel to the wrist-to-shoulder axis. Although this limits the number of possible arm poses available, it simplifies the complexity of inverse kinematic computations required to compute the ergonomic cost of each arm position. The elbow position can therefore be parameterised as a function of ϕ about the $\mathbf{\hat{u}}$ axis as:

$$e = r[\cos(\phi)\mathbf{\hat{u}} + \sin(\phi)\mathbf{\hat{v}}] + c \tag{4.1}$$

where r is the radius and c the centre of the circle traced by the swivelling elbow joint, and $\mathbf{\hat{u}}$ and $\mathbf{\hat{v}}$ are two unit vectors which form a local coordinate system for the plane containing the circle. After computation of the arm poses, we heuristically determine the ergonomic cost of regions in 3D space in terms of consumed endurance (CE), muscle reserve, and RULA using methods used by Belo et. al. [15]. For physical ergonomic cost calculations, we use data collected with the U.S. Army Anthropometric Survey (ANSUR) from 1987-1988 as used by Belo et. al [15]. While this method is efficient for real-time computations, fixing the wrist and shoulder positions reduces the number of reachable positions in the interaction space and produces motions where only six of the seven joint variables change between successive time steps [53]. Regardless, these changes are not noticeable in most cases for this application.

To simplify CE calculations for our UI toolkit, we use arm data for the 50th percentile male and compute CE for solely static poses. using methods described in [25]. The centre of mass (CoM) of a two segment body is located along the vector linking the CoMs of each segment, at a distance from the first segment's CoM equal to the ratio between the segment's mass and the combined masses of both segments [25]. Due to the fact that the hand coordinate is locked (always at zero degrees), the CoM of the elbow-hand vector will thus always be 17.25cm from the elbow for the 50th percentile male. Figure 4.1 displays the arm segments involved in computing its CoM, and Figure 4.2 shows the forces aggregated at the CoM.

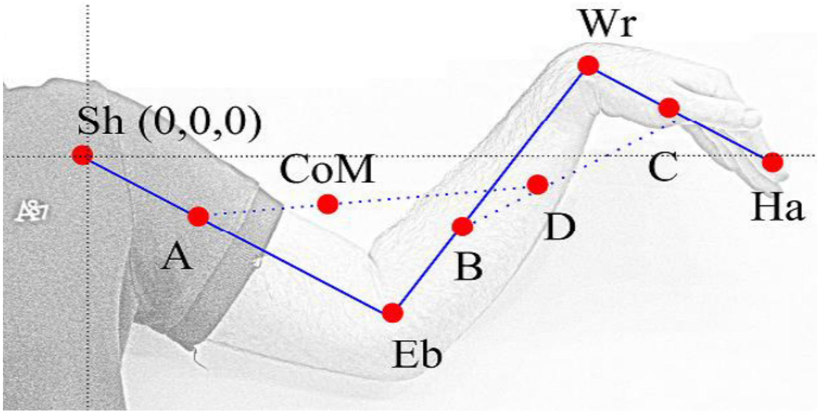


Fig. 4.1 Arm segments involved in calculating its CoM [25].

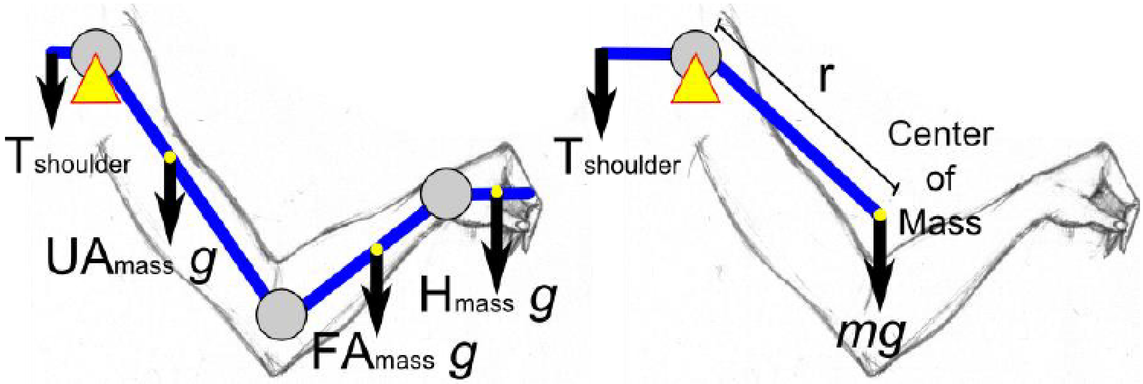


Fig. 4.2 (a) Primary forces acting on the upper arm, and (b) forces aggregated at the CoM [25].

The formula for the CoM of the forearm + hand combination can be written as:

$$D = B + \frac{WrHa_{mass}}{EbWr_{mass} + WrHa_{mass}} \overrightarrow{BC} \tag{4.2}$$

and the CoM of the upper arm + (forearm + hand) combination as:

$$CoM = A + \frac{EbWr_{mass} + WrHa_{mass}}{Arm_{mass}} \overrightarrow{AD} \tag{4.3}$$

where Eb is the elbow location, Wr is the wrist location, and Ha is the hand location as shown in Figure 4.1, and Wr_{mass} is the wrist mass, Ha_{mass} is the hand mass, and Arm_{mass} is the arm mass as shown in Figure 4.2. Using the arm mass of the 50th percentile male m, we then determine the force acting at the CoM as follows:

$$\overrightarrow{force_{motion,t}} = \overrightarrow{acceleration_t} * m \tag{4.4}$$

Therefore, the total torque of the system can be expressed as:

$$\sum \vec{T}_t = \vec{r} \times \overrightarrow{force}_{motion,t} \tag{4.5}$$

We derive the actual torque exerted by the shoulder muscles at time t:

$$\left\| \vec{T}_{shoulder,t} \right\| = \left\| \vec{r} \times \overrightarrow{force}_{motion,t} - (\vec{r} \times m\vec{g} + I_t \vec{\alpha}_t) \right\| \tag{4.6}$$

By representing the arm as a two segment body composed of upper arm, forearm, and hand, we may use the mathematical formulations of endurance provided above to study and guide the design of mid-air interactions. However, a more detailed analysis of CE modelling should include an extension of the model to capture other arm-segments, as well as usage of individual body metrics such as length and mass [25].

Next, we use static optimisation to estimate muscle activations for each voxel pose following Belo et. al. [15]. Each pose is analysed over time, and the timeframe which minimises the reserve actuation for each pose is saved. This results in an activation value for each muscle and reserve actuator in the model. These are combined into a single cost value by averaging the muscle activations and summing all the reserve actuators. Results which mostly use muscle forces are prioritised by penalising cases where reserve moments are high. Hence, we use the maximum reserve value of all voxels, where their reserve value is the minimum between all of the poses, as the threshold for the maximum acceptable reserve forces $T_{reserve}$. Voxels that have reserve values among $T_{reserve}$ receive the worst comfort rating. This results in the following cost function:

$$erg\ cost = \frac{\sum_{n=1}^{M} n_{activation}}{M} + \frac{\sum_{n=1}^{A} a_{activation}}{T_{reserve}} \tag{4.7}$$

where M is the number of muscles, A is the number of reserve actuators in the model, and $T_{reserve}$ is the threshold for the maximum acceptable reserve forces based on net joint moments.

Finally, we compute a RULA score for each arm pose according to scoring instructions for the upper arm, lower arm, and wrist [38]. First, the range of movements for the upper arm are assessed and scored as:

- 1 for 20 degree extension to 20 degrees of flexion
- 2 for extension greater than 20 degrees or 20 to 45 degrees of flexion
- 3 for 45 to 90 degrees of flexion
- 4 for 90 degrees or more of flexion

If the shoulder is elevated, the posture score derived as above is increased by 1. If the upper arm is abducted, the score is also increased by one. For the lower arm, the scores are:

- 1 for 60 to 100 degrees of flexion
- 2 for less than 60 or more than 100 degrees of flexion

If the lower arm is working across the midline of the body or out to the side, then the score is increased by 1. Due to a fixed neutral position, the scores for all voxel poses are incremented by 1.

As noted by McAtamney et. al [38], RULA is designed to provide a quick assessment of the loads on the musculoskeletal system of operators due to posture, muscle function and the forces they exert. Thus, while it may be used as a screening tool for assessing workplace safety, the metric should be combined with other models for a more robust assessment of fatigue and potential injury induced on the body.

For the CE, muscle activation, and RULA objective functions, the most ergonomically cost-efficient location to place a widget in the UI is defined as the voxel location which minimises the cost of these functions. We note that the use of these parameterised models reduces the number of poses in interaction space and only considers static poses of the arm. Nevertheless, this modelling provides a real-time, objective, non-invasive and non-obtrusive approach to assess the physical ergonomics of 3D UI interactions.

4.1.2 Cognitive Load

Due to the difficulty in quantifying the cognitive load of the user at a given instance, we enable the designer to manually specify the cognitive level through a slider widget in the Unity editor. The slider is scaled from 1 to 10, with 1 representing the lowest levels of cognitive load and 10 the highest levels. We adapt a similar method used by Lindlbauer et. al. [30] involving varying levels of detail (LOD) of the applications in the UI. In addition to the LODs, we create three sample 'virtual' real environments to replicate three different cognitive levels. These levels are described as follows:

- Low (1-3): The user is situated in a small office with a desk and seating area. UI widgets display the maximum level of content and are sized normally.
- Medium (4-6): The user is in a classroom setting with multiple rows of desks and chairs. Text and/or notifications displayed on UI widgets are truncated, and the size of the widgets are slightly reduced.
- High (7-10): The user is in a laboratory setting with equipment. UI widgets only display an icon with no notifications or text, and are downsized to a small cube.

Figures 4.3, 4.4, and 4.5 display these environments as they appear in Unity. Through this approach, the UI toolkit enables the designer to specify the level of detail and information displayed in widgets in context of potential environments the user may be in while interacting with such UIs.

Fig. 4.3 An office environment for a user with low cognitive load.

Fig. 4.4 A classroom environment for a user with medium cognitive load.

Fig. 4.5 A laboratory environment for a user with high cognitive load.

4.1.3 Target Acquisition

In terms of 3D UI design, we use Fitts' Law to describe the time taken by a user to touch an object displayed on the AR device screen. We design a simple Fitts' Law task using Unity, in which nine 10cm cubes are arranged in a vertical grid configuration in front of the user. During each iteration, a cube is chosen at random and coloured red to notify the user to touch the selected cube. Once the user selects the correct cube, another cube is chosen at random. The time taken for the user to select the correct cube is recorded for each iteration, as well as the distance from the centre point of the previous selected cube and the current selected cube. This was completed for 40 iterations. Figure 4.6 shows three configurations of the 3D Fitts' Law task in Unity, as well as an image of the hologram projection taken on the Hololens 2.

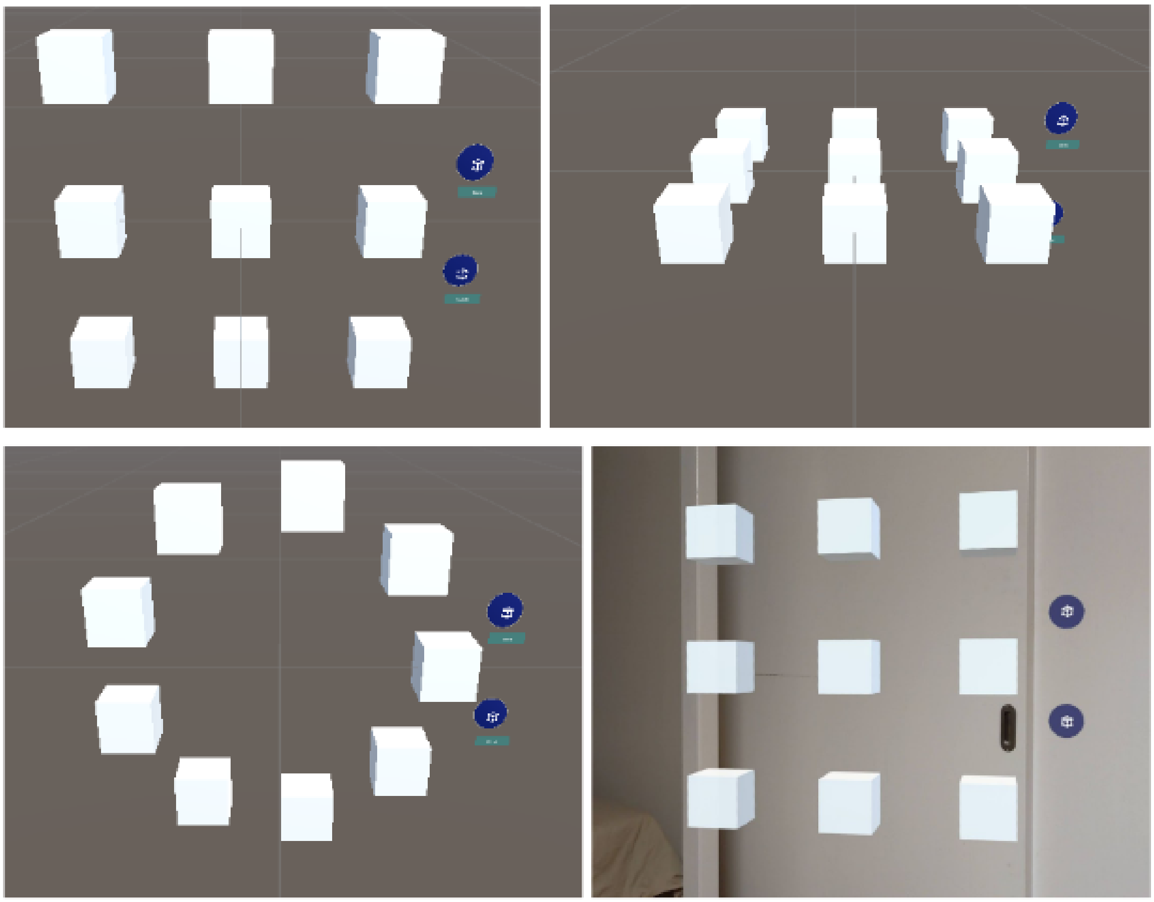


Fig. 4.6 Vertical grid, horizontal grid, and circular Fitts' Law task layouts which were designed using Unity. The bottom-right image displays the layout projected on the Hololens 2.

Using the distance and movement time values, we first compute the ID of each iteration of the task using Equation 3.1. We then plot ID against mean movement time (MT) in accordance to Equation 3.2 to empirically determine the values of a and b. Figure 4.7

displays a plot of ID vs MT, which yields a linear equation providing values of $a = 0.4926$ and $b = 0.6332$. We also conduct similar experiments for different configurations of the cubes, one in a horizontal grid layout and one in a circular layout as shown in Figure 4.6. When plotting ID vs MT for all three experiments in Figure 4.8, we observe no significant difference in values for a and b.

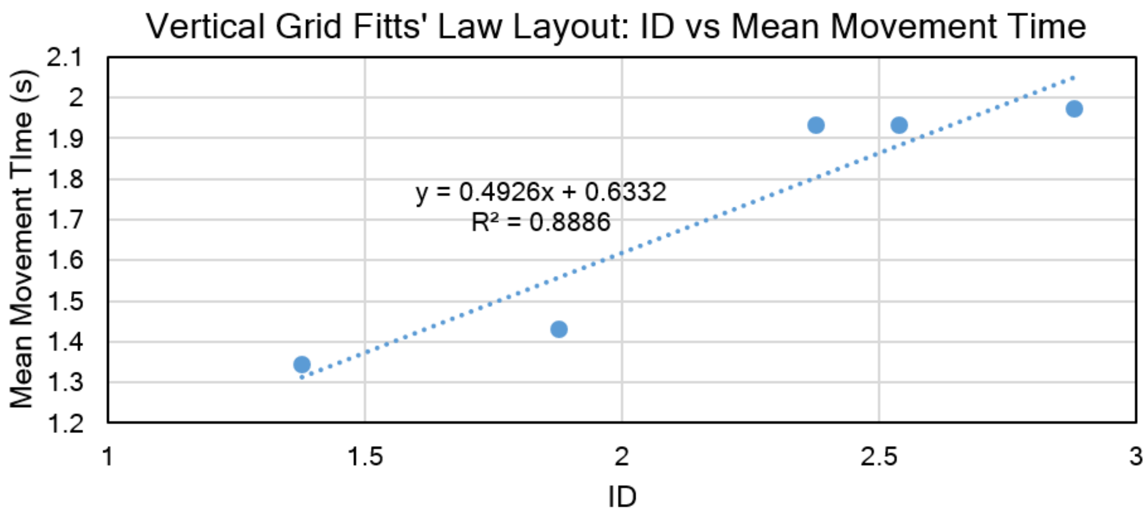


Fig. 4.7 Plot of ID vs MT for the vertical grid orientation 3D Fitts' Law task. The plot yields a linear equation of $MT = 0.4962ID + 0.6332$, corresponding to values of $a = 0.4926$ and $b = 0.6332$.

For our UI toolkit, we use values for a and b from the vertical grid layout to compute a score for the target acquisition objective function for each voxel. We describe A as the Euclidean distance between the current voxel position and the average centre point of widgets already positioned in the UI layout. W describes the width of the current widget to be placed in the UI. Voxel positions which minimise movement time are described to be most optimal in terms of target acquisition.

It is worth noting that the data used for formulation of a target acquisition objective function was taken from one user. By definition, Fitts' Law models the performance of humans; however, human performance is dependent on human traits and factors such as age, visual health, previous exposure to certain technologies, cognitive abilities, and so forth [54]. Thus, factors such as tiredness, concentration, and cognitive load may have an effect on user performance. For simplicity, we limit our target acquisition testing to one user and for one interaction (touch-press).

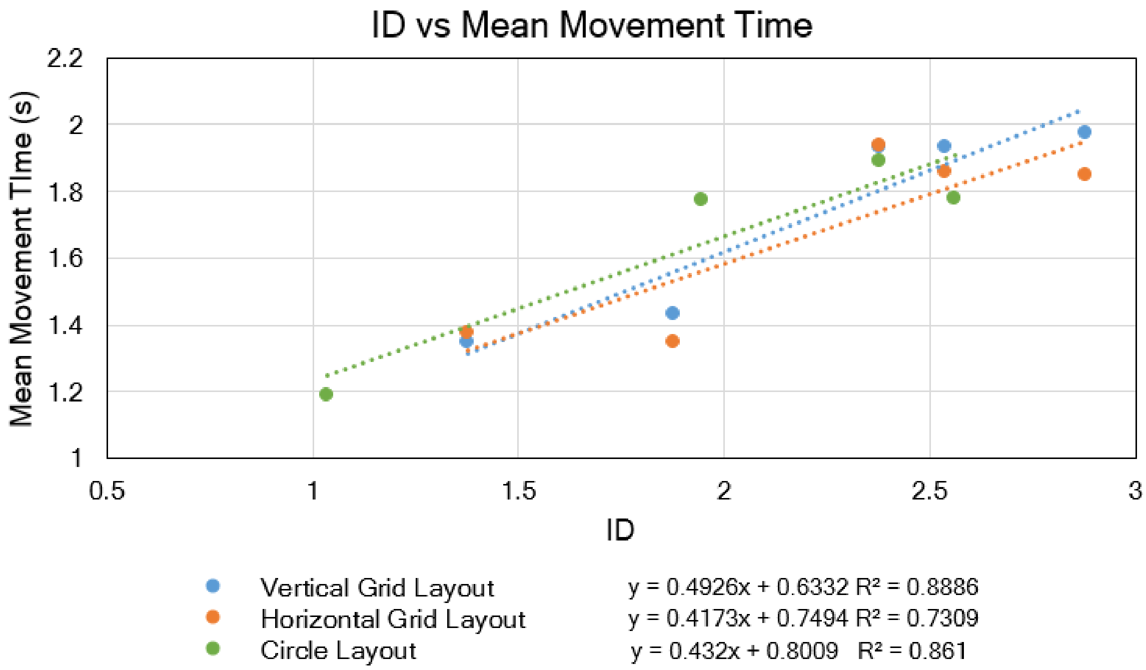


Fig. 4.8 Plots of *ID* vs *MT* for the vertical grid, horizontal grid, and circular orientation Fitts' Law tasks.

4.1.4 Text Legibility

Our toolkit operates under the assumption of a static user environment and thus requires an initial image of the users' environment. This initial image is used to compute and compare the colourfulness and edgeness of regions in the users' environment, which is also used in the determination of optimal placement of panels in terms of text legibility.

We adopt the method used by Dudley et. al. [14] in improving text legibility for text content for AR user interfaces. Using preference data collected through crowdsourcing, the authors score text label placement locations based on the texture and colouration of images taken of the users' background under the hypothesis that a highly colourful background region will be avoided when placing a label. The *colourfulness* metric M is based on an image's RGB colour space and is computed by first collapsing the colour channels as follows:

$$rg = R - G \tag{4.8}$$

$$yb = \frac{1}{2}(R+G) - B \tag{4.9}$$

where R, G, and B are the red, green, and blue bands of an image. These collapsed channels are then transformed into a representative mean μ_{rgyb} and standard deviation σ_{rgyb}:

$$\mu_{\mathrm{rgyb}} = \sqrt{\mu_{rg}^2 + \mu_{yb}^2} \tag{4.10}$$

$$\sigma_{rgyb} = \sqrt{\sigma_{rg}^2 + \sigma_{yb}^2} \tag{4.11}$$

The mean and standard deviation are finally used to compute M:

$$M = \sigma_{rgyb} + 0.3 \times \mu_{rgyb} \tag{4.12}$$

We also compute the *edgeness* per unit area F of an image, which quantifies the degree of texturing or 'busyness' of an image (CITE Dudley). The study defines the formula for F as:

$$F = \frac{\{|p|\mathrm{Mag}(p) \geq T|\}}{N} \tag{4.13}$$

or in other words, a score for a region of N pixels for which the count of the number of pixels p for which the gradient magnitude $\mathrm{Mag}(p)$ exceeds threshold T. Optimisation of the colourfulness and edgeness objective functions seeks the voxel position which yields the lowest M and F, respectively.

4.1.5 Colouration and Colour Harmony

When colouring the widgets in UI layouts, we integrate harmonic colour schemes developed by Matsuda [37] in the form of harmonic templates as described by Cohen-Or et. al. [11]. There are various shapes of harmonic templates, each of which specify a range of colours within 'wedges' defined on an HSV colour wheel as shown in Figure 3.1. These colours may consist of shades of the same colour, or shades which are complementary to one another. We choose to use the V-type template, which consists of a sector of 26 percent of an HSV colour wheel (or 93.6 degrees of a 360 degree wheel).

The UI toolkit first computes an initial colour for a specific widget based on the dominant colour and lightness of a patch in the initial image of the users' environment as conducted by Dudley et. al. [14]. We define a patch as the pixel region of the image corresponding to a certain voxel after the voxel's world coordinates have been converted into pixel coordinates. The dominant colour of a patch in the image c_p is extracted by taking the mode of the hue histogram and the mean of the saturation and value values in HSV space, while the lightness value of the patch l_p is grouped by thresholds on L* based on the CIE 1975 L*a*b* colour space. For each widget colour group g, we compute the probability of selecting g for the widget given c_p, $P(g|c_p)$, as well as the probability of selecting g given l_p, $P(g|l_p)$. We also find $P(g|\text{palette})$, the probability of selecting g given the defined colour palette. These three probabilities are combined to yield mixture distribution $G(g)$. The colour group g_{max} corresponding to the maximum of the mixture distribution $G(g)$ is then chosen. Finally, we choose the widget colour c_w corresponding to g_{max} in the palette, as well as the text colour based on the perceived brightness of colour c_w.

Using colour c_w and the HSV representation of c_w (c_{HSV}), the UI toolkit then chooses a random start and end angle θ_{start} and θ_{end} so that the following conditions are met:

$$|\theta_{\text{start}} - \theta_{\text{end}}| = 93.6^\circ \tag{4.14}$$

$$\theta_{\text{start}} < c_{HSV} < \theta_{\text{end}} \tag{4.15}$$

These conditions ensure that c_{HSV} falls within the angles defining the V-type template of 93.6 degrees of the HSV colour wheel. The toolkit generates colours within the range of these angles to produce a list of harmonious colours to c_w.

4.2 Objective Function Formulation

The general, the multi-objective optimisation (MOO) problem is given as follows:

$$\begin{aligned} \underset{\mathbf{x}}{\text{Minimize}} : \mathbf{F}(\mathbf{x}) &= [F_1(\mathbf{x}), F_2(\mathbf{x}), \cdots, F_k(\mathbf{x})]^T \\ \text{subject to } &: g_j(\mathbf{x}) \leq 0; j = 1, 2, \ldots, m \end{aligned} \tag{4.16}$$

where k is the number of objective functions and m is the number of inequality constraints, as explained by Marler et. al. [35]. $\mathbf{x} \in E^n$ is a vector of design variables, and $\mathbf{F}(\mathbf{x}) \in E^k$ is a vector of objective functions $F_i(\mathbf{x}) : E^n \to E^1$. The feasible design space is defined as $\mathbf{X} = \{\mathbf{x} \mid g_j(\mathbf{x}) \leq 0, j = 1, 2, \ldots, m\}$, and the feasible criterion space is defined as $\mathbf{Z} = \{\mathbf{F}(\mathbf{x}) \mid \mathbf{x} \in \mathbf{X}\}$.

The solution for the MOO problem can be unclear since a single point that minimises all objectives simultaneously usually does not exist. Therefore, the idea of Pareto optimality is used to describe solutions for such problems. A solution point is deemed Pareto optimal if it is not possible to move from that point and improve at least one objective function without worsening any other objective function. Typically, there are infinitely many Pareto optimal solutions for an MOO problem. Thus, it may be necessary to incorporate user preference for the objective functions in order to narrow down to a single optimal solution. For our UI toolkit, we choose to utilise an a priori articulation of preferences in which the user indicates their preferences prior to the optimisation process and allows the optimisation algorithm to determine a single solution which reflects these preferences. Specifically, we take a weighted sum optimisation approach as described by Marler et. al. [35] in which we seek to minimise a weighted combination of the outputs of the CE, muscle activation, RULA, target acquisition, colourfulness, and edgeness models:

$$U = \sum_{i=1}^{k} w_i F_i(\mathbf{x}) \tag{4.17}$$

where w_i is the scalar weight for each objective function, and $F_i(\mathbf{x})$ is the cost of each function. Because all of the weights must be positive, minimising the function for U provides a sufficient condition for Pareto optimality, which means the minimum of this function is always Pareto optimal [57, 20].

A benefit to this approach is that it transforms the original multi-objective optimisation problem into a single-objective optimisation problem. Thus, the solution methods for solving single-objective problems are all valid [35]. However, there are limitations to this approach. First, the method only works for convex Pareto fronts. Many authors demonstrate the method's inability to capture Pareto optimal points that lie on non-convex portions of the Pareto optimal curve [28][48][47][8] [2][26]. Second, it is usually difficult to generate a set of points which are uniformly distributed on the Pareto front. While these two limitations are well-documented, they concern the method's use for yielding a complete Pareto optimal set rather than a single solution (a priori articulation of preferences). Studies of the weighted sum method that focus on a priori articulation of preferences remain limited.

We choose to utilise the weighted sum method due to its simplicity. Our toolkit allows the designer to specify the weights w_i for each model using a slider scaled from 0 to 1. Using these weights, the UI toolkit then iterates through each voxel in the interaction space to determine the overall cost U associated with each voxel. The ergonomic costs (for the CE, muscle reserve, and RULA models) are based on the ergonomics of the arm pose required to reach a particular voxel. For the colourfulness and edgeness models, the toolkit converts the voxel position (which are in world coordinates) into pixel coordinates so that it may find the patch in the 2D environment image corresponding to each voxel. Once U has been computed for each voxel, the toolkit then chooses the voxel with the lowest U, v_{min}, and places the widget in the location of v_{min}. This process is repeated for each widget in the UI layout until the layout has been completely optimised.

4.3 Unity Toolkit

4.3.1 Instructions

Our UI toolkit is implemented in Unity, a cross-platform game engine commonly used for VR/AR application development. It operates under the assumption of a static user environment; in other words, the UI is always used in the same environment. The main steps for using the toolkit are as follows:

1. The designer connects an AR device to take an initial image of the users' environment using the webcam attached to the device. This will be the image used to construct the UI.

2. A script in the UI toolkit will send the image data through a socket connection from the AR device to the Unity server. This data contains the image bytes and metadata associated with the image (specifically, the Hololens camera to world matrix and projection matrix).

3. The image buffer file and metadata files obtained in the previous step are used as input to the toolkit. The designer will specify certain details of the UI they would like to design with the toolkit. For instance, they may specify the total number of widgets and the size and text for each widget. The designer may also use the sliders to tune the weights for each objective function which will be used in the optimisation process, and also select whether they would like the widgets to be colour-harmonised and if the widgets may/may not occlude others.

4. The toolkit then uses multi-objective weighted sum optimisation via a Python script to generate an optimal UI layout using the constraints specified by the designer. This design is then returned and shown in Unity. The designer may continue to use the sliders to adjust the weights of the objective functions, and the UI toolkit will continue to optimise and change the layout accordingly. Furthermore, the designer may add, remove, or change widgets. They may also specify a cognitive load to visualise the layout in different 'virtual' real environments and change the LOD of the UI.

4.3.2 Definitions

We define specific terms used in the toolkit as follows:

- **Colour Harmony**: Range of colours with similar hues on the HSV scale.

- **Colourfulness**: Measure based on the amount of colouration in the users' environment.
- **Edgeness**: Measure based on the amount of 'busyness' in the users' environment.
- **Fitts' Law**: Average movement time to each UI panel as a function of index of difficulty.
- **Consumed Endurance**: Severity of upper-arm fatigue from prolonged arm use.
- **Muscle Activation**: Muscle activation of the upper arms.
- **RULA**: Amount of 'risk' associated with the current arm posture.
- **Cognitive Load**: Measure of the users' workload or cognitive usage.

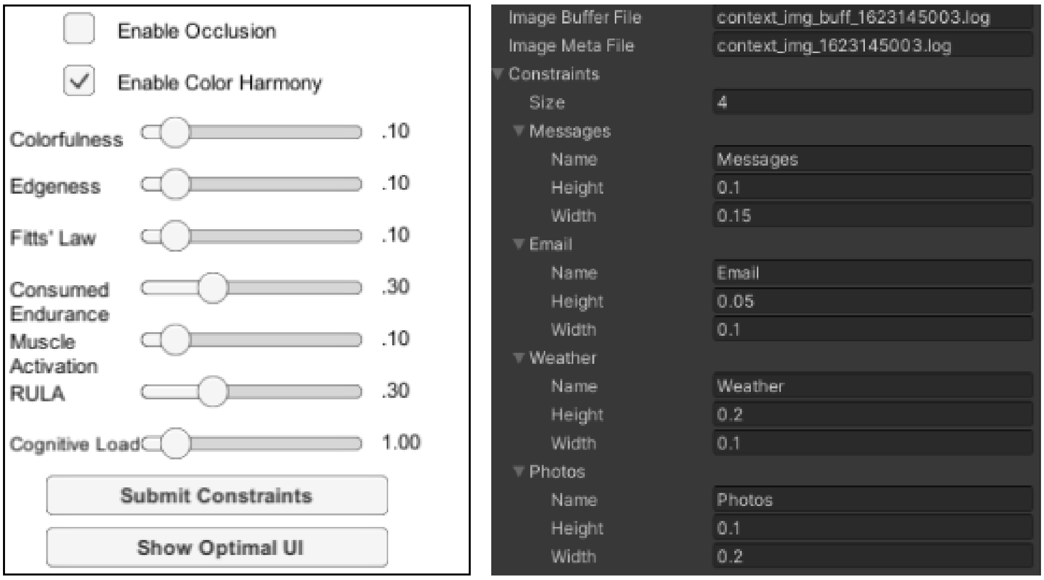
4.3.3 Designer Workflow

We describe our envisioned designer workflow as follows: The designer begins by adjusting the weights of the objective functions based on the factors which will be most impactful for the target user group. Once the UI is generated, they may further fine-tune the weights to continually adjust the UI layout. Once complete, the designer records the optimal locations and colours of the UI panels as well as the weights used. These can also be used in the preference learning application to adjust the UI layout based on preference learning.

Figure 4.9 (a) displays the toolkit's objective function menu with sliders to adjust the weights of each function. Figure 4.9 (b) shows the toolkit's constraint menu, which allows the designer to specify information for each widget in the UI. Figure 4.10 displays a sample UI generated in Unity as well as the widgets in the UI projected onto the original user environment image. A gallery of sample user environment images used for this project are included in Appendix A (and are also provided in the toolkit repository). Additionally, more UI layouts for different settings of weights are in the Appendix B.

4.4 Preference Learning Toolkit

In addition to the weighted sum multi-objective optimisation utilised by the Unity UI design toolkit, we also demonstrate usage of preference learning to yield an optimised UI layout. The preference learning application is a Python script integrated separately from the Unity toolkit, but still uses the user environment image captured with the Hololens as with the toolkit. We take a probabilistic kernel approach to preference learning based on Gaussian processes, as proposed by Chu and Ghahramani [9] in 2006. In this scenario, the pairwise preferences



(a) (b)

Fig. 4.9 (a) The toolkit's objective function menu, and (b) the constraints menu.

(a) (b)

Fig. 4.10 (a) A sample UI layout generated by the toolkit in Unity using the parameters and constraints picture in Figure 4.9, and (b) the UI layout generated onto the original user environment image.

between instances are the designer's preferences between two potential UI layouts. The method of capturing the designer's preference is summarised as follows:

1. Query the designer with a paired comparison between two UI designs and record the choice.
2. Update the Gaussian process model with the choice made by the designer.
3. Optimise a utility function which seeks a balance between exploration and exploitation of the latent function.

In our Python script implementation, we define X as training data consisting of numeric real positive values, and M as an array containing preferences. A preference is an array of positive integers, where the the left integer is the index of a value in X which is preferred over another value of X indexed by the right integer. Preference relations are captured in a Bayesian framework, allowing for global optimisation of the latent function values $f(x_i)$ describing each preference relation x_i. We use a Matern kernel with a length scale of 1 and $\mu = 2.5$ to specify the covariance function of the GP. The parameters for Laplace posterior approximation are noise = 0.0005, maximum number of iterations of 1000, gradient descent step size of 0.01, and gradient descent convergence tolerance of 0.0005. The acquisition function is the Upper Confidence Bound (UCB) function which enables an optimisation procedure to sample an optimal point based on attributes of the posterior distribution. The bounds of the optimisation space are set to the interaction space (in world coordinates) of the Unity toolkit. Figure 4.11 displays three sample iterations of the preference learning application: in the first iteration, the user chooses the suggested layout over the current layout. In the next iteration, the suggestion from the previous iteration becomes the preference, and the algorithm replaces the suggestion with a new layout. The user now chooses the preference layout over the new suggestion, so the preference layout remains the same in the third iteration.

Although the UI optimisation process is predominantly completed by the Unity UI toolkit, the preference learning toolkit serves the purpose of integrating the user-in-the-loop approach into our project. Figure 4.12 displays the overall UI optimisation process when both toolkits are combined. The constraints set by the designer (in terms of widget size, number, and type) are the inputs to the toolkit. These are used in Bayesian optimisation to generate an optimal UI layout in terms of the CE, muscle activation, RULA, target acquisition, colourfulness, and edgeness predictive model. Furthermore, the toolkit can utilise preference feedback from the designer to further refine and optimise the UI.

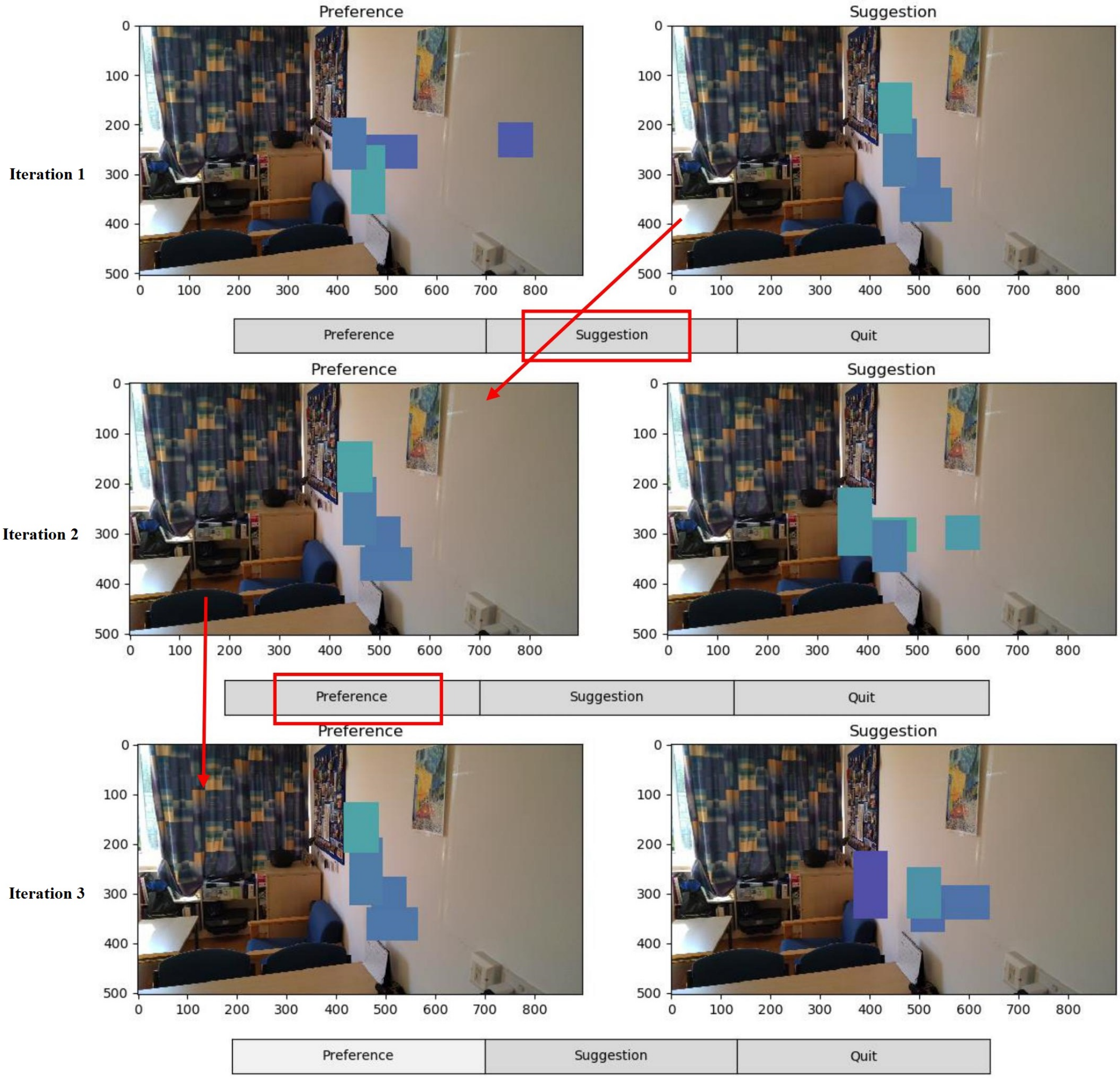


Fig. 4.11 Three sample iterations of the preference learning application. The user chooses one UI layout after being presented with a pairwise comparison between two layouts.

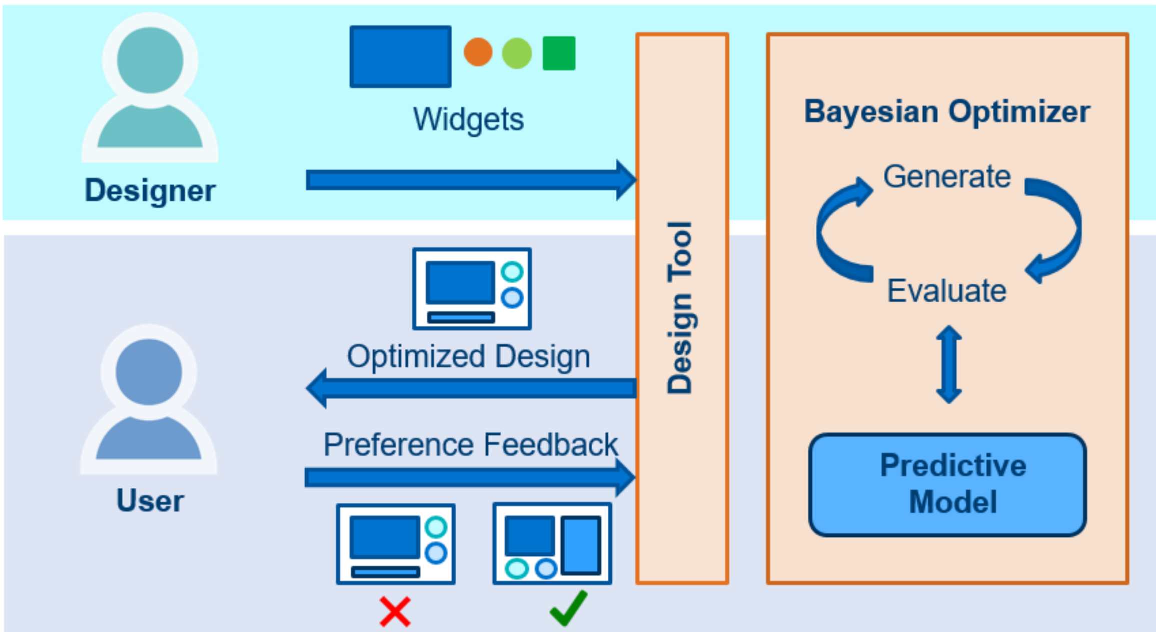


Fig. 4.12 Diagram of the UI optimisation process used by the UI design toolkit from a user-in-the-loop framework.

Chapter 5

Evaluation

In HCI, the *verification and validation* (V&V) process is an important step used to evaluate both the compliance of a system to specific requirements and its fulfilment of actual needs. The question of verification asks, “Does my system meet the specified requirements?", while the question of validation asks “Am I designing the correct product for operational needs?" The V&V process is often time-consuming and tedious, since it may involve a series of inspection, demonstration, test, and analysis cycles as well as usability inspections from real users. The process of *evaluation* is distinct, but often conflated with the V&V process. Evaluation is typically used to indicate success based on the ability to answer a research question, such as “Is method *A* better than *B* for some task?" In HCI, evaluation may verify that the hypotheses leading to a particular design has resulted in predicted beliefs.

5.1 Evaluation Goals and Errors

Before addressing our method of evaluating our UI design toolkit, we describe common goals for UI toolkit design as described by Olsen Jr. [42]:

1. **Reduce development viscosity**: The toolkit should reduce the time to create a new solution. In this case, our toolkit should allow the designer to create an inclusively immersive UI in a shorter amount of time than without using the toolkit.

2. **Least resistance to good solutions**: The toolkit should encapsulate and simplify expertise by utilising various optimisation methods as well as feedback from the designer.

3. **Lower skill barriers**: The toolkit should be simple and efficient to use, allowing designers with various skills and expertise levels to design UIs with ease.

4. **Power in common infrastructure**: The UIs designed with our toolkit should provide users with access to a multitude of abilities and services which would not have been easily available without using a UI.

5. **Enabling scale**: The variety and number of UI layouts constructed by our UI toolkit in a given amount of time should be greater than those available without using the toolkit.

Specific delineation of goals is useful for effective evaluation of systems, but we must also be aware of potential pitfalls to avoid. We must take careful consideration to avoid the verification-validation trap, a design process outcome in which a system passes verification but fails in validation. This is mainly attributed to the fact that the environment is incomplete when specifying the verification requirements, causing the environment to be inaccurately represented during the validation phase. To avoid this trap, it is useful to set realistic and specific verification targets. There are other potential pitfalls which may occur during the evaluation phase, as described by Olsen Jr. [42] as follows:

1. **Usability trap**: When testing the usability of an interactive system, researchers may mistakenly assume that all potential users may “walk up and use" a system. In other words, they assume that these users all have some basic knowledge on how to use this system, which would not work for those which require specialised expertise.

2. **Standardised task assumption**: A task designed for a usability experiment should have low inherent variability so that there are not a vast number of variables which may affect how a user approaches a task. For instance, a typing task for a text entry system would not have much variability in which it is approached. However, asking a user to sketch or design any UI may be ambiguous and subject to interpretation.

3. **Task scale**: Researchers often construct their usability tasks so that they may be completed in a short amount of time. However, more extensive testing would require significant amounts of time among a wider group of users, incurring a high cost.

Although the aforementioned scenarios are commonly understood to be adverse for usability testing, they are often challenging to avoid due to time and cost constraints. Nevertheless, understanding the goals and potential pitfalls which may occur during evaluation is still helpful for more effective system evaluation. We may now evaluate the UI toolkit through expert evaluation, which is detailed in the next subsection.

5.2 Expert Evaluation

We conduct expert evaluation, also known as heuristic evaluation, to evaluate the usability of our UI design toolkit. While typical evaluation methods for systems may involve usability studies with real users, time constraints and situational limitations imposed by the COVID-19 pandemic have limited our ability to test our toolkit with potential users. Furthermore, it would be infeasible to conduct remote user studies due to limited access to a Hololens device. Nevertheless, it is still possible for expert evaluators to approach a system through the users' perspective by going through the system process based on task scenarios.

5.2.1 Evaluation Goals

The two human experts in our evaluation possess prior experience with 3D UI layout design and AR technology. The evaluation of 3D UI layout designs is not an exact process, since these designs are subject to human preference and biases. Thus, to reduce the impact of subjective impressions, we define a set of criteria with which to evaluate the design toolkit. These criteria are based on Green's 'Cognitive Dimensions of Information Artefacts' [21], which serve as an aid in evaluating the usability of information-based artefacts. Information artefacts are tools used to store, manipulate, and display information such as word-processors, mobile devices, software environments, and so forth. The Cognitive Dimensions (CD) framework serves as a discussion tool to attain more simplified evaluations across a variety of criteria, rather than detailed analyses. We define the relevant CD criteria as follows:

1. **Viscosity**: Resistance to change; the cost of making small changes. In terms of the UI toolkit, this considers how sensitive the positioning of the UI components are to the adjusting of objective function weights.

2. **Premature Commitment**: The amount to which a designer must make some decisions (submitting UI layout constraints) prior to when proper information (the layout) is available.

3. **Progressive Evaluation**: The work in progress (the layout) can easily be checked during the designing process.

4. **Role-Expressiveness**: How well the purpose of each component (the various sliders, checkboxes, and input fields in the toolkit) can be inferred.

5. **Visibility**: Ability to view components of the toolkit easily.

6. **Error-Proneness**: The toolkit invites mistakes.

In addition to the cognitive definitions defined previously, the following two factors are also considered during the evaluation process:

1. **Sensitivity (sensitivity analysis)**: The experts determine the sensitivity in dependent variables (the positioning of widgets in the UI layout) when manipulating the independent variables (the weights of the various objective functions).
2. **Validity (model validation)**: The experts test the ability of the toolkit to create effective UIs for new user groups or groups with different traits than those tested previously.

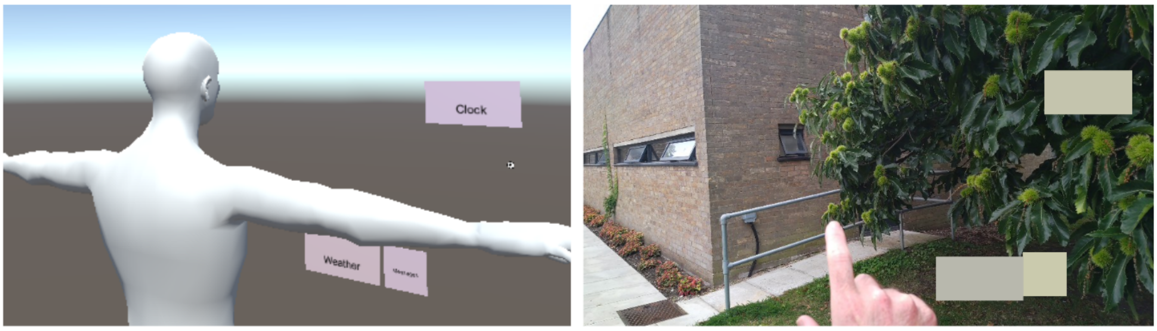
5.2.2 Procedure

For the evaluation, we construct four design tasks for each expert evaluator to complete:

1. Design UI layouts for users with limited upper body mobility (e.g. for users who may be interacting with other items in their physical environment or those with a physical disability).
2. Design UI layouts for users in a cluttered physical environment (e.g. for users who are in an environment with many people/objects such as a classroom, library, or laboratory).
3. Design UI layouts using preferences from the designer.
4. Design UI layouts for an outdoor environment.

After initial training and familiarisation with the toolkit, the evaluators are asked to construct a design for each task in a time frame of 1.5 hours for all four tasks. At the end of the task, they assign ratings from a scale of 1-10 (where 1 represents a design which does not meet the task specifications, and 10 is a design which completely meets specifications) to their saved designs and also to the other expert's designs based on how well they believe the design meets the task's specifications. Figure 5.1 displays a sample design created by Evaluator 1 for Task 4 after using the Hololens to take an image outdoors.

Once all the tasks are finished, we ask the designers to evaluate the toolkit as a whole based on the CD criteria listed previously on a scale of 1-10 for each criteria (where 1 is a negative evaluation and 10 is a positive evaluation according to the specific criteria). Finally, we gather further information about the usability and efficiency of the toolkit through a questionnaire and semi-structured interview.



(a) (b)

Fig. 5.1 A design created for Design Task 4 by Expert Evaluator 1. (a) The UI layout design in Unity, and (b) the layout generated onto the original user environment image.

5.2.3 Results and Feedback

Self ratings and secondary expert ratings for each UI layout designed during the four tasks are provided in Table 5.1. Each evaluator also assessed the Unity UI toolkit based on the six CD criteria. Results are recorded in Table 5.2.

	Design Task	Self Rating	Secondary Expert Rating
Expert Evaluator 1	1	7/10	6/10
	2	8/10	8/10
	3	7/10	8/10
	4	6/10	7/10
Expert Evaluator 2	1	6/10	6/10
	2	7/10	7/10
	3	7/10	7/10
	4	7/10	7/10

Table 5.1 Ratings given by each evaluator and secondary evaluator to each UI layout created for each design task.

After completion of the design tasks, each evaluator provided comments and feedback pertaining to the validity, limitations, and potential extensions of the toolkit. Evaluator 1 noted that there was some difficulty in enforcing separation between panels in the layout, especially when the weights for CE, Fitts' Law, and muscle activation objective functions were increased. Furthermore, there should be a tighter integration of components within the toolkit. Specifically, the integration of the preference learning application and the Unity UI toolkit would be fundamental in improving the robustness and usability of the toolkit. Nevertheless, Evaluator 1 mentioned that the toolkit helps to avoid a degree of design fixation and also visualise potential alternatives that they would otherwise not consider.

	Expert Evaluator 1	Expert Evaluator 2
Viscosity	6/10	6/10
Premature Commitment	5/10	5/10
Progressive Evaluation	6/10	7/10
Role-Expressiveness	7/10	7/10
Visibility	7/10	6/10
Error-Proneness	8/10	7/10

Table 5.2 Unity UI toolkit scores given by each evaluator based on the six relevant Cognitive Dimensions of Notation criteria.

Evaluator 2 also agreed with the comment that the separation between the preference learning application and Unity UI toolkit was a limiting factor for the overall design toolkit. Ideally, the toolkit would display a gallery of layout suggestions (as shown in the preference learning application) within Unity to allow the designer to explore other potential options. Evaluator 2 also mentioned several other ideas for improvement. For instance, displaying the previous UI layout designed in the Unity UI toolkit, along with the current layout, would be useful for comparing the two layouts. Furthermore, a “go back" or “undo" button could allow the designer to revert back to the previous design generated if they were unsatisfied with the current design. The evaluator also mentioned several benefits: first, the ability to visualise their design in different ‘virtual’ real environments was helpful in simulating potential contexts in which their UI layout may be used. In addition, recording the optimised panel location outputs and function weights from the toolkit enables easy replication of previously created layouts.

Although we were unable to extend our evaluation to more expert evaluators and/or potential users, both expert evaluators were able to inspect the compliance of the toolkit to specific requirements (by rating the toolkit based on CD criteria) and its fulfilment of actual needs (by rating UI layout designs created to meet specific design tasks). This evaluation method, while simpler in nature, has demonstrated the ability to complete an efficient V&V and evaluation of a system in a shorter period of time.

Chapter 6

Discussion

6.1 Design Implications

This project has shown four main implications: first, it is possible to identify and analyse parameters affecting human performance and comfort when interacting with 3D UIs through parameter analysis. The factor which affect the user experience in VR and AR interaction are not only limited to physical ergonomics, but also extend to visual ergonomics and cognitive capabilities. Furthermore, such factors are dynamic and subject to change; for example, a user's cognitive load may vary depending on the task at hand or the environment they are in. By conducting envelope analyses of these parameters, we have seen that it is possible to find the best settings for optimisation towards the designer's objectives.

Second, this project has demonstrated that it is possible to convert these parameters into quantitative objective functions and also transform multiple objective functions into a single objective function via a weighted sum optimisation process. Furthermore, we have demonstrated that it is possible to utilise a model-based approach from design engineering to validate systems which are complex and costly in the absence of data from actual or proxy users.

Third, we have shown the ability to use preference learning to integrate the users' feedback into UI optimisation in a process known as the user-in-the-loop approach. Due to noisy behaviour from users and variability between user preferences in terms of human perception, psychology, and preference, the UI optimisation process is non-trivial. Nevertheless, we have seen that capturing the users' preferences through a Bayesian framework enables optimisation of a utility function which seeks a balance between exploration and exploitation of the latent function.

Finally, the project has demonstrated that expert evaluation is a method which can be used to qualitatively evaluate a system prototype. By creating and conducting several design tasks and outlining specific criteria pertaining to the cognitive dimensions of notation, we have gained a better understanding of the usability, benefits, and limitations of our UI design toolkit.

6.2 Limitations

During the construction and evaluation of the UI design toolkit, we have noted a few limitations of the toolkit. These limitations are predominantly due to the inability to evaluate the toolkit with a greater number of designers, which is necessary for a robust evaluation process. Due to time constraints and limitations imposed by the COVID-19 pandemic, it was not possible to evaluate the UI toolkit with a greater number of users. Other potential methods may be in the form of crowdsourcing or conducting user surveys; however, both methods would require a significant amount of time to be taken for recruiting potential users and creating the survey or design tasks. However, it would be difficult to recruit users due to limited access to a Hololens device. Furthermore, these methods may require compensation in terms of the difficulty and/or time consumption of the tasks. Ideally, we would recruit several designers to complete a few design tasks, which may involve designing UIs using the UI design toolkit for specific applications or target users. The toolkit would then be evaluated based on effectiveness of the designers to achieve these tasks, as well as their own opinions of the usability and efficiency of the toolkit.

Another limitation is the simplicity of the toolkit. This UI toolkit is used to create menu-type UIs which would allow the user to select an application to launch. However, it would be useful for the designer to design different forms of UIs. For instance, an informational GUI display may show websites or instructional content which could be used in an educational setting. Additionally, a notification GUI could show alerts or messages when the user is preoccupied with other tasks.

Furthermore, our UI toolkit operates under several assumptions. First, it assumes that the only interaction type between the user and the UI widgets is a touch-press. Realistically, UIs designed for VR and AR devices allow for a variety of interactions such as dragging and pointing. However, enabling multiple interactions would require more extensive target acquisition tasks to obtain parameter values for these interactions.

Additionally, the toolkit assumes that the users' body is static. In order to run in real-time, the calculations for CE, muscle activation, and RULA only consider static arm poses. However, this simplification in inverse kinematic calculations limits the number of possible poses represented by the model. Additionally, ignoring motion would not be realistic for UI applications which require more user interaction and movements of the arms and upper body.

This trade-off was made due to the consideration that existing models that analyse movement and fatigue would be difficult to incorporate, especially for real-time computations.

A final limitation is the separation of the Unity UI toolkit and the preference learning toolkit, as mentioned by both expert evaluators. It would be more beneficial if the preference learning application of the project were in the form of a preference gallery displaying various potential UI layouts within Unity. This would allow the designer to better visualise many potential UI layouts in the context of the 'virtual' real environments.

6.3 Future Work

There are many potential extensions to this project given more time and resources. The integration of the preference learning toolkit into Unity would be a significant improvement, since this would allow for a smoother user experience for the designer. Furthermore, we would like for the designer to be able to construct a wider variety of UI types, such as a menu, notification, or informational UI. Another extension may be to create canonical user profiles with different attributes which would impact how the UI will render. For example, one potential profile may be a mechanic in-training who is using a UI to instruct him or her on how to fix a certain part of a vehicle. Another may be an individual with cerebral palsy with limited movement and/or coordination using a UI to learn in a classroom environment.

We may also make improvements to our current objective functions or integrate other models to our UI toolkit. For instance, improvements to inverse kinematic computations made in the physical ergonomics models could yield higher accuracy and also allow for more arm poses. Belo et. al. [15] suggests extending the kinematic chain to consider the wrist angle and using a spiral point algorithm at each voxel to compute the possible wrist positions. Other models could be integrated as well to consider other human traits such as colour blindness or visual impairments.

Another improvement would be the creation of adaptable UI layouts which allow for dynamic user environments. For example, the user may move to another location or be in a setting where objects and/or other people in their environment are moving. Utilisation of real-time visual and spatial information from the users' environment from the AR device could allow the layouts to continually adjust to changes in scenery, objects, or locations.

A final extension to this project would be conducting more extensive validation. It would be beneficial to allow other designers to test and evaluate the toolkit, and also rate its efficiency in the form of interviews or surveys.

Chapter 7

Conclusion

This project has successfully met its two objectives. First, we have explored various controllable and uncontrollable parameters which dominate user performance and comfort when interacting with VR and AR user interfaces. This was conducted via a model-based approach, in which design knowledge in the form of user simulations, models, and/or heuristics are used to model how users interact and perceive UI layouts. We have shown that physical ergonomic models (CE, muscle activation, and RULA), perceptual models (colour harmony and text legibility), and cognitive models (cognitive load) affect the users' experience. From the completion of this first object, we have demonstrated the use of a model-based approach in constructing a cost function which may be optimised via Bayesian optimisation and preference learning.

Second, we have designed and implemented a UI design toolkit for constructing 3D UI layouts which can suggest alternative configurations pertaining to user capability to the designer at design time. The construction of a model-based UI design toolkit has revealed potential in integrating techniques adapted from design engineering, especially for systems which are complex and costly to validate. By allowing the designer to adjust the objective function weights in multi-objective weighted optimisation, the designer can continuously adjust the optimised UI layout in terms of the controllable parameters explored previously. Furthermore, the integration of Bayesian optimisation and preference learning enables the designer to visualise multiple variations of their UI designs from the perspective of the user in 'virtual' real environments.

The combination of parameter analysis and the construction of the UI design toolkit has aided in understanding how to improve the accessibility of VR and AR systems for users with varying degrees of perceptual, cognitive, and physical capabilities. Rather than taking a universal design approach, which aims to develop systems for general use with a 'one size fits all' mentality, we have developed our toolkit through ability-based design [56]. Our project has demonstrated that utilising an ability-based design perspective is beneficial for focusing on ability throughout the design process and can create systems which leverage the full range of human potential. In the future, we aim to refine our current function models to enable more accurate and efficient creation of UIs, as well as explore and integrate other models to adapt to other user capabilities.

References

[1] Albarelli, A., Celentano, A., Cosmo, L., and Marchi, R. (2015). On the interplay between data overlay and real-world context using see-through displays. In *Proceedings of the 11th Biannual Conference on Italian SIGCHI Chapter*, CHItaly 2015, page 58–65, New York, NY, USA. Association for Computing Machinery.

[2] Athan, T. W. and Papalambros, P. Y. (1996). A note on weighted criteria methods for compromise solutions in multi-objective optimization. *Engineering Optimization*, 27(2):155–176.

[3] Bachynskyi, M., Palmas, G., Oulasvirta, A., and Weinkauf, T. (2015). Informing the design of novel input methods with muscle coactivation clustering. *ACM Trans. Comput.-Hum. Interact.*, 21(6).

[4] Beatty, J. and Lucero-Wagoner, B. (2000). *The pupillary system in Handbook of psychophysiology*. Cambridge University Press.

[5] Blandford, A. (2013). *Semi-structured qualitative studies*.

[6] Brochu, E., Brochu, T., and de Freitas, N. (2010). A bayesian interactive optimization approach to procedural animation design. SCA '10, Goslar, DEU. Eurographics Association.

[7] Chang, S., Dai, P., Hong, L., Sheng, C., Zhang, T., and Chi, E. H. (2016). Appgrouper: Knowledge-based interactive clustering tool for app search results. In *Proceedings of the 21st International Conference on Intelligent User Interfaces*, IUI '16, page 348–358, New York, NY, USA. Association for Computing Machinery.

[8] Chen, W., Wiecek, M., and Zhang, J. (1999). Quality utility : a compromise programming approach to robust design. *Journal of Mechanical Design*, 121:179–187.

[9] Chu, W. and Ghahramani, Z. (2005). Preference learning with gaussian processes. In *Proceedings of the 22nd International Conference on Machine Learning*, ICML '05, page 137–144, New York, NY, USA. Association for Computing Machinery.

[10] Clark, L. D., Bhagat, A. B., and Riggs, S. L. (2020). Extending fitts' law in three-dimensional virtual environments with current low-cost virtual reality technology. *International Journal of Human-Computer Studies*, 139:102413.

[11] Cohen-Or, D., Sorkine, O., Gal, R., Leyvand, T., and Xu, Y.-Q. (2006). Color harmonization. *ACM Trans. Graph.*, 25(3):624–630.

[12] Delp, S., Anderson, F., Arnold, A., Loan, P., Habib, A., John, C., Guendelman, E., and Thelen, D. (2007). Opensim: Open-source software to create and analyze dynamic simulations of movement. *Biomedical Engineering, IEEE Transactions on*, 54:1940 – 1950.

[13] Dias, S., Diniz, J., Konstantinidis, E., Savvidis, T., Zilidou, V., Bamidis, P., Grammatikopoulou, A., Dimitropoulos, K., Grammalidis, N., Jaeger, H., Stadtschnitzer, M., Silva, H., Telo, G., Ioakeimidis, I., Ntakakis, G., Karayiannis, F., Huchet, E., Hoermann, V., Filis, K., Theodoropoulou, E., Lyberopoulos, G., Kyritsis, K., Papadopoulos, A., Depoulos, A., Trivedi, D., Chaudhuri, R., Klingelhoefer, L., Reichmann, H., Bostantzopoulou, S., Katsarou, Z., Iakovakis, D., Hadjidimitriou, S., Charisis, V., Apostolidis, G., and Hadjileontiadis, L. (2020). Assistive hci-serious games co-design insights: The case study of i-prognosis personalized game suite for parkinson's disease. *Frontiers in Psychology*, 11.

[14] Dudley, J. J., Jacques, J. T., and Kristensson, P. O. (2021). *Crowdsourcing Design Guidance for Contextual Adaptation of Text Content in Augmented Reality*. Association for Computing Machinery, New York, NY, USA.

[15] Evangelista Belo, J. a. M., Feit, A. M., Feuchtner, T., and Grønbæk, K. (2021). *XR-gonomics: Facilitating the Creation of Ergonomic 3D Interfaces*. Association for Computing Machinery, New York, NY, USA.

[16] Evers, C., Kniewel, R., Geihs, K., and Schmidt, L. (2014). The user in the loop: Enabling user participation for self-adaptive applications. *Future Gener. Comput. Syst.*, 34:110–123.

[17] Fitts, P. (1954). The information capacity of the human motor system in controlling the amplitude of movement. *Journal of experimental psychology*, 47 6:381–391.

[18] Flatla, D. R., Reinecke, K., Gutwin, C., and Gajos, K. Z. (2013). *SPRWeb: Preserving Subjective Responses to Website Colour Schemes through Automatic Recolouring*, page 2069–2078. Association for Computing Machinery, New York, NY, USA.

[19] Gajos, K. Z., Weld, D. S., and Wobbrock, J. O. (2010). Automatically generating personalized user interfaces with supple. *Artificial Intelligence*, 174(12):910–950.

[20] Goicoechea, A., Hansen, D., and Duckstein, L. (1982). *Multiobjective decision analysis with engineering and business application*. John Wiley and Sons Inc., New York, NY, USA.

[21] Green, T. and Blackwell, A. (1998). Cognitive dimensions of information artefacts: a tutorial.

[22] Hart, S. G. and Staveland, L. E. (1988). Development of nasa-tlx (task load index): Results of empirical and theoretical research. In Hancock, P. A. and Meshkati, N., editors, *Human Mental Workload*, volume 52 of *Advances in Psychology*, pages 139–183. North-Holland.

[23] Hasler, D. and Suesstrunk, S. (2003). Measuring colourfulness in natural images. *Proceedings of SPIE - The International Society for Optical Engineering*, 5007:87–95.

[24] Hess, E. H. and Polt, J. M. (1964). Pupil size in relation to mental activity during simple problem-solving. *Science*, 143(3611):1190–1192.

[25] Hincapié-Ramos, J. D., Guo, X., Moghadasian, P., and Irani, P. (2014). Consumed endurance: A metric to quantify arm fatigue of mid-air interactions. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '14, page 1063–1072, New York, NY, USA. Association for Computing Machinery.

[26] Huang C-H, Galuski J, B. C. (2007). Multi-objective pareto concurrent subspace optimization for multidisciplinary design. *AIAA J*, 45:1894–1906.

[27] Itten, J. (1973). *The Art of Color: the Subjective Experience and Objective Rationale of Color*. John Wiley.

[28] Koski, J. (1985). Defectiveness of weighting method in multicriterion optimization of structures. *Communications in Applied Numerical Methods*, 1(6):333–337.

[29] Kristensson, P. O., Lilley, J., Black, R., and Waller, A. (2020). *A Design Engineering Approach for Quantitatively Exploring Context-Aware Sentence Retrieval for Nonspeaking Individuals with Motor Disabilities*, page 1–11. Association for Computing Machinery, New York, NY, USA.

[30] Lindlbauer, D., Feit, A. M., and Hilliges, O. (2019). Context-aware online adaptation of mixed reality interfaces. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology*, UIST '19, page 147–160, New York, NY, USA. Association for Computing Machinery.

[31] Losing, V., Hammer, B., and Wersing, H. (2015). Interactive online learning for obstacle classification on a mobile robot. In *2015 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8.

[32] MacKenzie, I. S. (1992). Fitts' law as a research and design tool in human-computer interaction. *Hum.-Comput. Interact.*, 7(1):91–139.

[33] MacKenzie, I. S. (2013). A note on the information-theoretic basis for fitts' law. *Journal of Motor Behavior*, 21:323–330.

[34] Manghisi, V. M., Gattullo, M., Fiorentino, M., Uva, A. E., Marino, F., Bevilacqua, V., and Monno, G. (2017). Predicting text legibility over textured digital backgrounds for a monocular optical see-through display. *Presence: Teleoper. Virtual Environ.*, 26(1):1–15.

[35] Marler, R. and Arora, J. (2010). The weighted sum method for multi-objective optimization: new insights. *Structural and Multidisciplinary Optimization*, 41:853–862.

[36] Mathur, A., Gehrmann, J., and Atchison, D. (2013). Pupil shape as viewed along the horizontal visual field. *Journal of vision*, 13.

[37] Matsuda, Y. (1995). *Color design*. Asakura Shoten.

[38] McAtamney, L. and Nigel Corlett, E. (1993). Rula: a survey method for the investigation of work-related upper limb disorders. *Applied Ergonomics*, 24(2):91–99.

[39] Moon, P. and Spencer, D. E. (1943). A metric based on the composite color stimulus. *J. Opt. Soc. Am.*, 33(5):270–277.

[40] Moon, P. and Spencer, D. E. (1944). Geometric formulation of classical color harmony∗. *J. Opt. Soc. Am.*, 34(1):46–59.

[41] Mott, M., Tang, J., Kane, S., Cutrell, E., and Morris, M. R. (2020). "i just went into it assuming that i wouldn't be able to have the full experience": Understanding the accessibility of virtual reality for people with limited mobility. In *ASSETS 2020*. ACM.

[42] Olsen Jr, D. (2007). Evaluating user interface systems research. pages 251–258.

[43] Rasmussen, C. E. (004). *Gaussian Processes in Machine Learning*, pages 63–71. Springer Berlin Heidelberg, Berlin, Heidelberg.

[44] Saul, K., Hu, X., Goehler, C., Vidt, M., Daly, M., Velisar, A., and Murray, W. (2014). Benchmarking of dynamic simulation predictions in two software platforms using an upper limb musculoskeletal model. *Computer methods in biomechanics and biomedical engineering*, 18:1–14.

[45] Shahriari, B., Swersky, K., Wang, Z., Adams, R. P., and de Freitas, N. (2016). Taking the human out of the loop: A review of bayesian optimization. *Proceedings of the IEEE*, 104(1):148–175.

[46] Snoek, J., Larochelle, H., and Adams, R. P. (2012). Practical bayesian optimization of machine learning algorithms.

[47] Stadler, W. (1995). Caveats and boons of multicriteria optimization. *Computer-aided Civil and Infrastructure Engineering*, 10:291–299.

[48] Stadler W, D. J. (1992). *Multicriteria optimization in engineering: a tutorial and survey*. American Institute of Aeronautics and Astronautics, Washington D.C., USA.

[49] Stockman, G. and Shapiro, L. G. (2001). *Computer Vision*. Prentice Hall PTR, USA, 1st edition.

[50] Sweller, J. (1994). Cognitive load theory, learning difficulty, and instructional design. *Learning and Instruction*, 4(4):295–312.

[51] Tanuwidjaja, E., Huynh, D., Koa, K., Nguyen, C., Shao, C., Torbett, P., Emmenegger, C., and Weibel, N. (2014). Chroma: A wearable augmented-reality solution for color blindness. In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, UbiComp '14, page 799–810, New York, NY, USA. Association for Computing Machinery.

[52] Todi, K., Weir, D., and Oulasvirta, A. (2016). Sketchplore: Sketch and explore with a layout optimiser. pages 543–555.

[53] Tolani, D. and Badler, N. (1996). Real-time inverse kinematics of the human arm. *Presence (Cambridge, Mass.)*, 5:393–401.

[54] Triantafyllidis, E. and Li, Z. (2021). The challenges in modeling human performance in 3d space with fitts' law. *CoRR*, abs/2101.00260.

[55] van der Heijden, J. (2003). Ubiquitous computing, user control, and user performance: conceptual model and preliminary experimental design. In Lechner, U., editor, *Research Symposium on Emerging Electronic Markets*.

[56] Wobbrock, J. O., Kane, S. K., Gajos, K. Z., Harada, S., and Froehlich, J. (2011). Ability-based design: Concept, principles and examples. *ACM Trans. Access. Comput.*, 3(3).

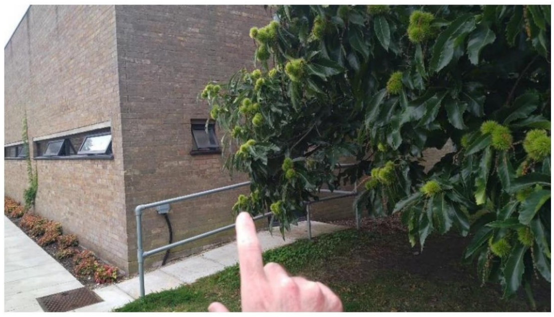
[57] Zadeh, L. (1963). Optimality and non-scalar-valued performance criteria. *IEEE Transactions on Automatic Control*, 8:59–60.

Appendix A

Gallery of Sample Environment Images

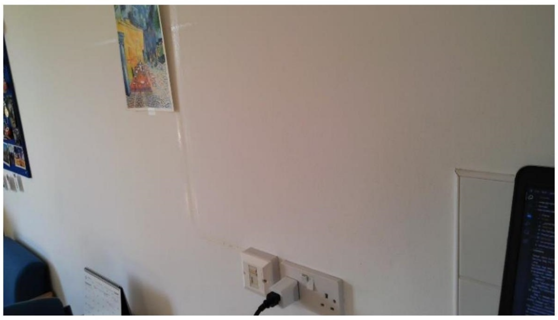
The following images in Figures A.1 and A.2 were taken with the Hololens 2 and used as the input environment images when generating optimal UI layouts. All of the following images are included in the Unity Toolkit repository.

context_img_1629213409.png



context_img_1629213950.png

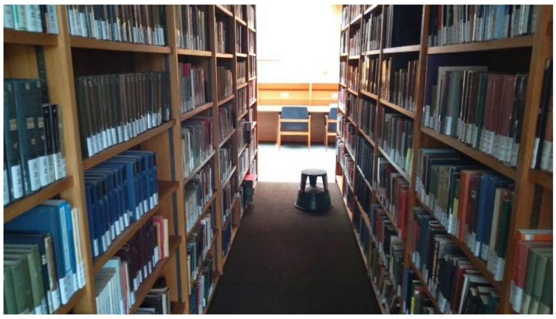
Fig. A.1 Sample user environment images taken in various outdoor locations.



context_img_1623145003.png

context_img_1629133512.png

context_img_1629197228.png



context_img_1629197347.png

context_img_1629197515.png

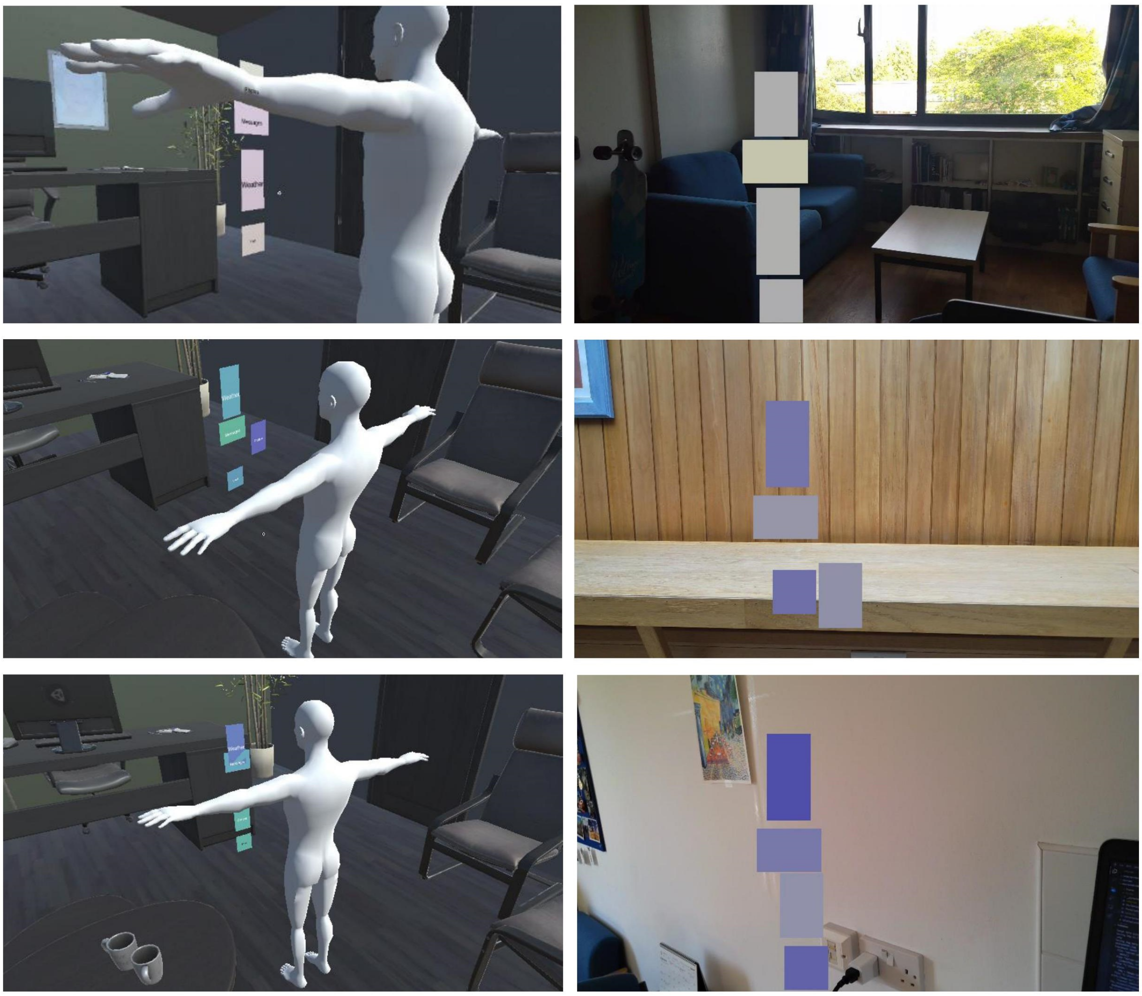
context_img_1629197623.png

Fig. A.2 Sample user environment images taken in various indoor locations.

Appendix B

Gallery of Sample UI Layouts

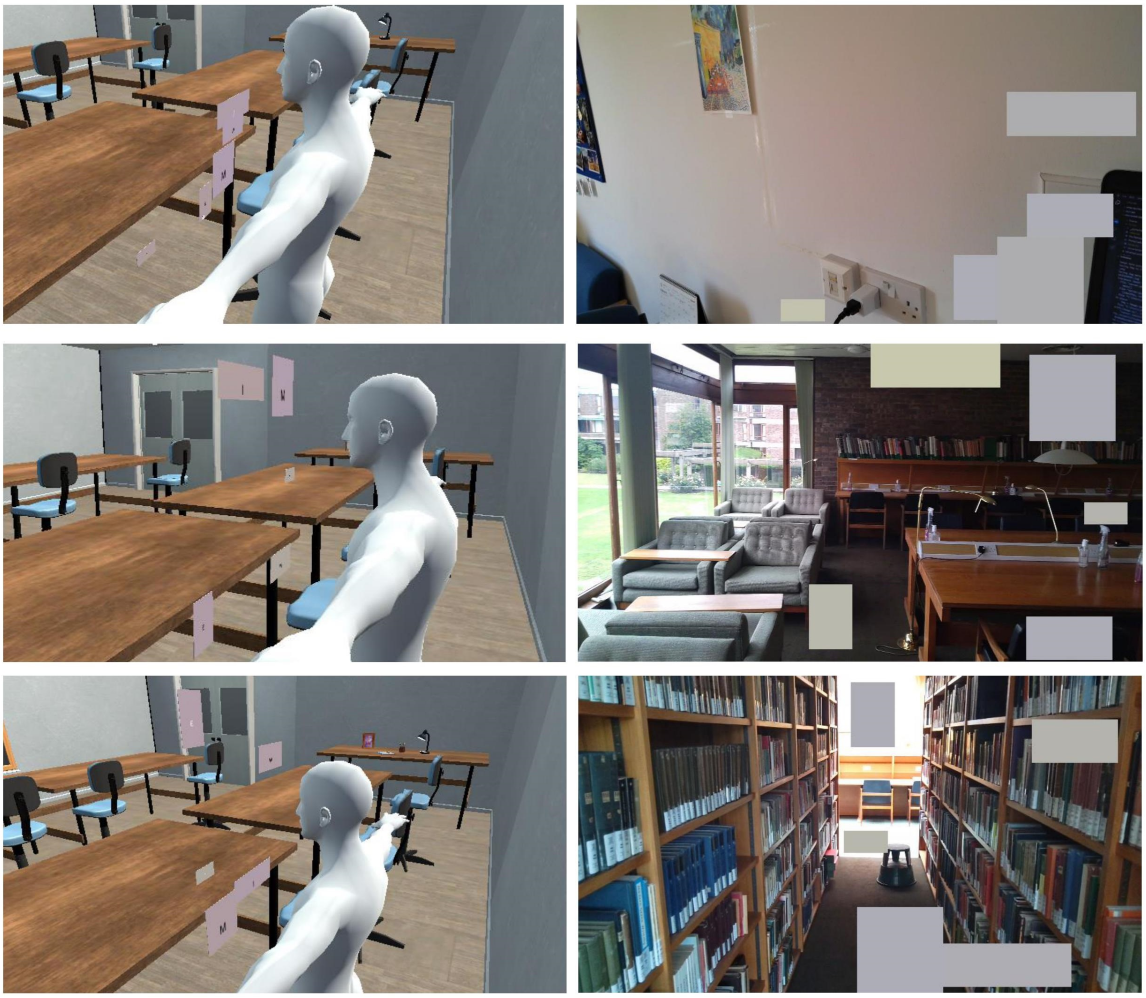
The following UI layouts were generated with the provided objective function weights and constraints. ‘Enable occlusion’ is set to false and ‘Enable colour harmonisation’ is set to true for all layouts. If a weight is not listed, then it is set to 0. The left image displays the optimised UI layout in Unity, and the right image displays the layout on the reference environment image used.



$w_{CE} = w_{MuscleActivation} = w_{RULA} = 0.33$, cognitive load = 1

Panel Dimensions = [(0.1, 0.15), (0.1, 0.1), (0.2, 0.1), (0.15, 0.1)]

Fig. B.1 Sample UI layouts generated from various environment images for the set of weights listed.



$w_{Colorfulness} = 0.75, w_{Edgeness} = 0.15, w_{Fitts'Law} = 0.1$, cognitive load = 6

Panel Dimensions = [(0.2, 0.2), (0.15, 0.1), (0.1, 0.2), (0.05, 0.1), (0.1, 0.3)]

Fig. B.2 Sample UI layouts generated from various environment images for the set of weights listed.

$w_{Edgeness} = 0.5, w_{Fitts'Law} = 0.25, w_{RULA} = 0.25$, cognitive load = 9

Panel Dimensions = [(0.2, 0.2), (0.3, 0.3), (0.1, 0.2)]

Fig. B.3 Sample UI layouts generated from various environment images for the set of weights listed.

Appendix C

Source Code

- Source code for the Unity UI Toolkit is available at: https://github.com/jwlee97/UnityUIToolkit
- Source code for the Preference Learning application is available at: https://github.com/jwlee97/PreferenceLearningApplication