Abstract:

The excitement surrounding the virtual reality industry is spreading across all sectors of society. Nevertheless, the highest projected market value is expected to be in entertainment and gaming. The advancements in the visual, auditory, and kinesthetic departments have been impressive and are well known. However, the endeavors that attempt to incorporate the tactile perception have been undermined and do not seem to enjoy the same popularity.

We, the development team (the Team) aim to develop Spellbound, a two-player virtual reality game tailored with strategic and first-person-shooter (FPS) elements. Furthermore, the Team will also develop two haptic vests to highlight how integrating haptic feedback can result in an enjoyable gaming experience.
## Version Control

<table>
<thead>
<tr>
<th>Date</th>
<th>Editor</th>
<th>Change Log</th>
<th>Version Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>24 September 2017</td>
<td>Sheng Leung</td>
<td>Plan Draft</td>
<td>1.0</td>
</tr>
<tr>
<td></td>
<td>Cheung Casado</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1 October, 2017</td>
<td>Sheng Leung</td>
<td>First Deliverable</td>
<td>2.0</td>
</tr>
<tr>
<td></td>
<td>Cheung Casado</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Table of Contents

1. Introduction ................................................................................................. 4
  1.1. History ..................................................................................................... 4
  1.2. Related Works ........................................................................................ 4

2. Problem Statement ...................................................................................... 6

3. Project Expectations .................................................................................. 7
  3.1. Scope ....................................................................................................... 7
    3.1.1. Locomotion Detection ....................................................................... 7
    3.1.2. Haptic Feedback Hardware ............................................................... 7
    3.1.3. Game Application ............................................................................. 7
  3.2. Objectives ................................................................................................. 7
  3.3. Deliverables ............................................................................................. 8
    3.3.1. Phase I .............................................................................................. 8
    3.3.2. Phase II ............................................................................................ 8

4. Prerequisites ............................................................................................... 9
  4.1. Hardware .................................................................................................. 9
    4.1.1. Virtual Reality Head-Mounted Devices (HMD) .................................. 9
    4.1.2. Supporting Computer(s) .................................................................... 9
    4.1.3. Leap Motion ....................................................................................... 9
    4.1.4. Microsoft Kinect ............................................................................... 9
    4.1.5. Arduino Uno & Relevant Components .............................................. 10
  4.2. Software ................................................................................................... 10
    4.2.1. Unity ................................................................................................ 10
    4.2.2. Blender .............................................................................................. 10
    4.2.3. Image Editor ..................................................................................... 10
  4.3. Scripting ................................................................................................... 11
    4.3.1. C# .................................................................................................... 11
    4.3.2. C/C++ .............................................................................................. 11

5. Methodology .............................................................................................. 12
  5.1. Software Development Practice ............................................................... 12
  5.2. System Overview .................................................................................... 12
  5.3. Game Development ................................................................................ 13
    5.3.1. Game Flow & Mechanics ................................................................ 13
    5.3.2. User Interface Design ................................................................. 14
  5.4. Haptic Suit Development ....................................................................... 16

6. Feasibility & Risk Assessment ................................................................... 17
  6.1. Main Concern & Mitigation Strategies .................................................... 17
  6.2. Schedule .................................................................................................. 17

7. Conclusions ............................................................................................... 19

8. References ................................................................................................ 20
1. Introduction

Multiple market projections forecast that the virtual, augmented, and mixed reality applications (VAMR applications) field will become the next multibillion-dollar industry within a decade from now [1, 2, 3]. The advent of applications such as Pokémon Go! [4], hardware such as the iPhone X’s TrueDepth camera [5], and the multiple growing communities of Oculus, HTC Vive, or Microsoft Hololens users, can attest to such claims as they exemplify that the prevalence of VAMR apps is becoming increasingly apparent [6, 7]. Virtual reality (VR) is defined as a computer generated 3D environment, which can be interacted by a person [8], and is expected to become a big part of the entertainment and gaming industry [1]. VR devices appeal heavily to the senses of the users, which helps to create an immersive experience. Haptic devices, for example, are those that simulate the perception of touch and can also be used to enhance the VR experience [9].

The structure of this paper is as follows. The remainder of this section covers details regarding the history of VR and certain kinds of VR devices. It also describes relevant efforts in haptic device research. Section two states the motivation driving this project. Section three defines the project scope, lists the project objectives, and states the expected deliverables. Section four is the prerequisites section, which explains the design decision rationale of hardware and software relevant to the project methodology. Section five is the project methodology, which describes the development cycle ideology, an overview of the system architecture, and also the game software and hardware development. Section six is the feasibility and risk assessment, which also includes the schedule and pointers on quality assurance.

1.1. History

Although research efforts in visual, auditory, kinesthetic, and even haptic VR technology stemmed as early as the late 80s to early 90s, it was not until the 21st century that it gained public recognition when investors such as Facebook, Google, Samsung, Sony, Microsoft, HTC etc. started to invest in this technology. Among the types of VR devices, VR headsets quickly became famous. 2014 saw the $2 billion acquisition of Oculus VR by Facebook, the introduction of Google Cardboard and the announcement of the Sony PlayStation VR [10]. Around the same period, there were several haptic devices technology endeavors, but despite the technology outburst, these never seemed to gain similar traction [11].

1.2. Related Works

In 2010, students at the University of Pennsylvania developed the Tactile Gaming Vest (TGV), which enables users to play Half-Life 2 and feel haptic feedback similar to being shot, or slashed. Though reportedly, the vest only responded to receiving game events. No feedback would be provided if the game event originated from the user of the vest [12]. A Kickstarter effort in 2013 titled As Real As It Gets (ARAIG) attempted to expand on the same concept but in addition to mechanically generated haptic feedback, the ARAIG product aimed to also use electrical muscle stimulation
to emulate certain touch perceptions. The ARAIG group also recorded a demo in which their vest was capable for providing haptic feedback for a game event that originated from the wearer of the vest (e.g. shot recoil) [13]. However, neither vest was integrated to VR equipment as users played the video game using a desktop computer [11,12].

The *Aura Interactor* was one of the earlier haptic gaming devices. Commercialized in 1994, it was the first of its kind to use electromagnetic actuators to convert bass sound waves into vibrations [14]. This idea of adding the haptic sensation due strictly to auditory game events is experiencing a reemergence. *Woojer*, a 2017 startup, is currently manufacturing a vest with improved technology along with a SDK for developers to provide a service similar to what the *Aura Interactor* achieved [15]. The competitor to *Woojer* is *KOR-FX*, which originated from a successful *Kickstarter* campaign in 2014. The *KOR-FX* vest has already been tested with recent games like *Resident Evil 7* and is commercially available [16].

The *Hardlight VR Suit* is under development by *NullSpace VR Inc.* and aims to provide both vibratory feedback similar to the TGV and the audio-generated vibratory sensations akin to the *Woojer* and *KOR-FX* products [17]. Contrary to *Hardlight VR*, which is targeted to be easily affordable and for home use, companies like *The Void* have opened a handful of theme-parks that provide a similar experience, albeit more costly, by using their product, the *Rapture VR* set: a combo of a VR head-mounted display (HMD) and a haptic vest [18]. Lastly, the *Teslasuit*, currently under development by *Tesla Studios*, is one of the most ambitious projects in haptic feedback VR devices. It aims to be the first haptic feedback solution that is full-body, provides an array of haptic feedback sensations, has motion capture and positioning systems built-in, is thermo-regulated, and is capable of multiplayer [19].
2. Problem Statement

There are two issues with the existing efforts in haptic feedback VR games: Firstly, there is a difference in focus. The majority of commercially available options for haptic feedback do not offer support for VR games, and those that claim to do so are still under development [20]. Secondly, there is a general lack of haptic feedback multiplayer VR games (according to steam search results as of the time of this writing).

The current focus for frontrunners in haptic VR devices is on providing the tactile experience as a whole package to maximize the commercial value of the technology. Consequently, not only are developers implementing the haptic sensation, but also valuable development time is being consumed building technologies to layer atop the haptic feedback functionality (e.g. independent position tracking, and gesture recognition).

Although creating such APIs provide some long-term benefits in terms of the standardization in the future development process, it is also important to expose the haptic-enabled VR experience to the public quickly to explore potential short-term unknowns. For example: medical concerns analogous to those caused by VR headsets could arise [21]; alternatively, the public may simply disapprove of the device design or find the gaming experience too cumbersome. In the case of multiplayer VR games that aim to include a haptic component, it is especially worth it to clarify such prospects since they are few in number.

Thus, it is convenient and faster to integrate proven technologies such as Leap Motion and Kinect to investigate such possibilities.
3. Project Expectations

3.1. Scope

The scope of this project is divided into three parts:

3.1.1. Locomotion Detection

Capturing player movements is an essential component in any VR application. The corresponding technologies (see Prerequisites) will be appropriately integrated to capture body movement, gesture, orientation, and position as input for the haptic feedback vest or the game application. At this stage the estimated play-space required is undetermined since the Kinect performance needs to be tested under the expected usage conditions.

3.1.2. Haptic Feedback Hardware

Two haptic vests shall be built. The types of haptic feedback to be provided shall be mechanical in nature. The Team shall test vibration intensities, frequencies, and locations to best emulate different types of tactile sensations. The current scope of the project does not include the audio-to-mechanical conversion features (i.e. services provided by Woojer and KOR-FX, see Related Works).

3.1.3. Game Application

A two-player strategy game in which one user combats the other user with some FPS elements will be implemented (see Methodologies, Game Development). Since the current scope does not include story, and quest components, the Team believes that adding multiple characters for the players to experiment with will be beneficial to the gaming experience. However, this is highly dependent on the progress of the Team during the development of the network component of this project (see Feasibility & Risk Assessment, Main Concerns & Mitigation Strategies).

3.2. Objectives

- To develop, Spellbound a two-player virtual reality game that contains strategic and FPS elements; and,
- To integrate Leap Motion technology to allow for game interactions through gesture recognition; and,
- To develop two haptic vests capable of providing basic tactile feedback relative to a game event (e.g. getting shot, or shot recoil); and,
- To integrate Microsoft Kinect technology to enable haptic feedback based on the position of a game event (e.g. getting shot from the side results in tactile sensation on the side of the vest).
3.3. Deliverables

We have organized the development process into two phases. Below is a list of expected, tangible deliverables:

3.3.1. Phase I

- Basic character model
- Physically defined playing area
- Framework for game with core game mechanics
- Haptic suit prototype

3.3.2. Phase II

- Network component
- Two finalized haptic suits
- Game demo with tutorial
4. Prerequisites

4.1. Hardware

4.1.1. Virtual Reality Head-Mounted Devices (HMD)

Choosing the correct VR hardware is paramount to realize the development of our game. The two main contenders are the Oculus Rift, or the HTC Vive and while the spec comparison is close, the key difference is in the tracking systems. The Oculus uses a Constellation tracking camera, whereas the Vive uses a lighthouse laser tracking system. The issue with the Constellation tracking camera is that the maximum tracking space is eight by eight, any larger and occlusion may occur. In comparison, the lighthouse laser system is capable of a maximum tracking space of 15 by 15 feet [22].

4.1.2. Supporting Computer(s)

While the Vive provides impressive performance, it also has hefty demands to maintain 90fps gameplay. The specific requirements are computer running on Windows 7.1 or later with either an Nvidia GTX 1060, or AMD RX 480 graphics card, and an AMD FX 8350 or an Intel i5tr-4590 processor with at least 4GB of RAM. At this stage of development, we predict that more than one computer will likely be necessary to support the multiplayer experience [22].

4.1.3. Leap Motion

The Leap Motion device is a renowned vision-based gesture-tracking device that uses an infrared pattern projection to define a hover zone. Within this area the technology is capable of recognizing the hands of the user. Furthermore, while the device has some gestures that it is able to identify by default, it is also possible to program unique gestures [23].

Another incentive to use leap motion is its modest dimensions and weight, so it can be easily attached to the Vive HMD without hindering the playing experience.

4.1.4. Microsoft Kinect

The Kinect is a 3D motion-sensing camera, which also uses infrared technology. Though the Leap Motion device is capable of motion detection, it is highly specialized for hand tracking, motion and gestures. In comparison, the Kinect is able to do the same for the entire body (i.e. skeleton tracking), which is necessary to realize the game mechanics [24].

The Kinect SDK has support for Unity, and Unity also provides packages to aid in general development [24].

There have been studies regarding the positional accuracy and performance of the Kinect [9], the Team shall account for such issues during development.
4.1.5. Arduino Uno & Relevant Components

The Arduino Uno is a single-board microcontroller. Among the options for microcontrollers, the Arduino Uno is well supported by Unity in the form of the plugin Uniduino [25, 26]. Also, the Arduino products was the only option we found to have their own IDE, which we expect is likely to simplify coding.

Along with the microcontroller will be the auxiliary wiring and other pieces that will recreate the haptic sensations upon signaling. As stated in the scope section, the Team will first focus on recreating mechanically induced haptic sensations, depending on the development process the possibility of including other haptic sensations may be entertained. To accomplish this vision, the necessary transducers, actuators, or vibrators will also be required. However, currently the Team has yet to review the specifics regarding these additional components.

4.2. Software

4.2.1. Unity

Unity is a free game engine that supports an array of VR plug-ins. Not only is the engine powerful, but also its development environment is very versatile. Animations, audio, game objects, the user interface, etc. Unity provides a multitude of tools to tailor and facilitate development of these elements and more [27].

The Team is also more familiar with Unity development, its documentation, and its community more than other well-reputed game engines such as Unreal.

4.2.2. Blender

Blender is an open source program for 3D content-creation. It has multiple modules for animating and rigging, and it also supports VR rendering. The main incentive for using Blender is that there are free-licensed projects that have been tested in Unity, enabling us to spend less time designing and focusing more on the game mechanics implementation, and haptic suit realization [28].

4.2.3. Image Editor

Adobe Photoshop and Gimp are the image editing applications we may use when creating game assets, if the asset requires more specialized editing than what is provided by the Unity development environment.
4.3. Scripting

4.3.1. C#

The programming languages supported by Unity are C# and JavaScript. The popular opinion on the Unity forums is that it is considered better practice to use C# as the language reinforces good coding principles. There have also been some tests that suggest the engine performs better on C# scripting. The official documentation is generally better supported for C# than JavaScript as well [27].

4.3.2. C/C++

Technical prowess in C/C++ is required since the Arduino language is wrapped around C/C++. That is, the methods called in the Arduino IDE are essentially C/C++ functions [25].
5. Methodology

5.1. Software Development Practice

Given the limited timeframe and the small number of developers, the most suitable software development methodology is the Agile methodology, which is appropriate for volatile requirements and rapid changes in the development process or environment [29]. Therefore, in accordance to Agile principles, we organized development into smaller milestones to achieve in weekly iterations, which will be evaluated at the beginning and end of every week.

More about our intended approach is outlined on the Feasibility & Risk Assessment, Schedule section.

5.2. System Overview

![Figure 1](https://via.placeholder.com/150)

**Figure 1** Architecture Overview of System for *Spellbound*. Under motion capture the three devices (from top to bottom) are: Microsoft Kinect, Leap Motion, HTC Vive.

The above figure (see Figure 1) outlines the general idea of the project where:

- The straight lines to the Unity Engine block connects the relevant SDK, libraries, and plug-ins that enable the devices to communicate, implying that said modules are integrated.
- The pointed arrows represent the transfer of input to or from the supporting computer running Unity.
  - The Kinect shall provide information about the skeleton of the users.

Cheung Casado, Sheng Leung 3035123863
Joshi, Abhinav 3035203857

Page 12 of 22
• The Leap Motion device shall do the same, but exclusively for the hands.
• The HTC Vive HMD sends information about its orientation to the supporting computer, which in turn replies with data about how to display the 3D visuals.
• The microprocessor on the haptic device receives signals to turn on the component to simulate a tactile stimulus and also indicates when it has finished.

5.3. Game Development

5.3.1. Game Flow & Mechanics

Figure 2.1 The different phases in Spellbound.

Figure 2.2 Rock-paper-scissors nature of the game mechanics in Spellbound

After two players are successfully paired, they will be place on opposing sides of the middle of one of five lanes on the virtual gameplay environment. The game would proceed as depicted in Figure 2.1.

Select phase: During the Select Phase, players interact with a virtual menu to choose their desired action for this turn. The actions include: move, attack, grab, and defend. In addition to choosing an action, the player must also target a lane. For example, if player selects move, then the player also must choose which lane to switch to on his side of the field. Alternatively, if the player chooses to attack, then the player must select the lane on which the opponent may be in.
**Evaluation phase:** During the *Evaluation Phase*, the game decides what will occur during the *Performance Phase* (see Figure 2.2). The relationship between attack, grab, and defend is similar to rock-paper-scissors, but only resolves if the selected actions involve the same lane.

**Performance phase:** During the *Performance Phase*, the player has to execute his action via gestures. For example, if a player chooses attack and it is the correct lane, then during this phase the player would have to take aim and perform a gesture to damage the opponent. It is the interactions during this exchange that will signal the tactile feedback on the haptic suit.

**Resolution phase:** During this phase additional game events that relate to the character the player chooses are resolved. For example, the ammunition for the move option is replenished, or the character heals some damage.

5.3.2. User Interface Design

![Menu UI](image_url)

Figure 3.1 Menu UI

The UI for the game consists of a virtual menu and the HUD as depicted in the figures 3.1, 3.2, and 3.3 respectively.

**Menu:** The Player encounters the menu as soon as the game boots up. The Menu (see Figure 3.1) is rendered inside the Virtual environment which may consist of any of the maps present in the game.

The player has the following options to choose from:

- New game: Starts up a new instance of the game and then searches for a pair. On locating it pairs up and starts the game.
- Character Selection: Prompts the player to choose between different characters
- Settings: Prompts the player to choose between different Audio/Video/Game settings the player would like to change.
- View models: The player can view the 3d models created in the game
- View Maps: the player can view and choose between different maps in the game
HUD (Heads up display): The player once connected into the game has a visual feedback system which enable the player to view the status as well as choose between options provided to him (see Figure 3.2). Example: The health bar, the actions available, etc.

The following items are visible to the user:

- Health Bar: A percentage meter which provides the user with His/her health information.
- Action choice: Provides the Player the available action he/she can perform. Example Attack, Defend, Grab and Move.
- Hint prompts: Provides the player with Hints/Options. It can be from asking to choose between actions or providing the player with new Inputs like “Damage Taken”, “loosing Health”, “Grabbed potion Health Increased”, etc.
- Movement arrows: Helps the Player to look and check available places to move.

Figure 3.2 HUD overlay

Figure 3.3 Example UI during gameplay

The above figure (see Figure 3.3) is an example of the expected UI during the Selection Phase as described earlier on the Game Flow & Mechanics section.
5.4. Haptic Suit Development

At this stage, the design for the haptic vest has yet to be finalized. The points that still need to be clarified are:

- Size of the haptic suit
- Number of types of tactile feedback to emulate
- Number of locations that apply the effect

The following step is to map the devices that are wired to the microcontroller to the character model on the Unity platform.
6. Feasibility & Risk Assessment

6.1. Main Concern & Mitigation Strategies

The primary concern for the Team is the networking component of the software. We fear that the game concept may be implemented poorly in terms of performance, and when run through a network that it may ruin the user experience. Testing the latency of the microcontroller, and the motion capture technologies will definitely be an area of focus. Depending on whether it is a hardware issue, or a coding issue, additional research will be carried out to either improve the hardware setup or the runtime performance for the scripts. If independent research is insufficient, we may seek for an expert opinion through our advisor, or other personal connections.

Another concern is quality assurance particularly and user experience. To achieve this we plan to recruit three types of testers: those without gaming experience; those with gaming experience but have not used VR equipment; and, those with gaming experience that also uses VR equipment. We also plan to ask participants for user experience feedback in the form of a questionnaire.

6.2. Schedule

*Note: The milestones are listed on the Deliverables section*

<table>
<thead>
<tr>
<th>Week</th>
<th>Dates</th>
<th>Tasks</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1st – 7th</td>
<td>- Finalize design choices for the:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>o Haptic device</td>
</tr>
<tr>
<td></td>
<td></td>
<td>o UI of each game phase</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Inspect playing field size</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Equipment acquisition/booking</td>
</tr>
<tr>
<td>2</td>
<td>8th – 14th</td>
<td>- Motion capture environment setup/integration</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Character design and basic player model</td>
</tr>
<tr>
<td>3</td>
<td>15th – 21st</td>
<td>- Character design</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Map design</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Game flow implementation</td>
</tr>
<tr>
<td></td>
<td></td>
<td>o Basic game object layout for different game flow phases</td>
</tr>
<tr>
<td></td>
<td></td>
<td>o Basic UI</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Arduino setup</td>
</tr>
<tr>
<td>4</td>
<td>22nd – 28th</td>
<td>- Game flow implementation</td>
</tr>
<tr>
<td></td>
<td></td>
<td>o Character select</td>
</tr>
<tr>
<td></td>
<td></td>
<td>o Map select</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Game mechanics</td>
</tr>
<tr>
<td></td>
<td></td>
<td>o Default &amp; Custom Gestures</td>
</tr>
<tr>
<td></td>
<td></td>
<td>o Movement</td>
</tr>
<tr>
<td></td>
<td></td>
<td>o Health</td>
</tr>
<tr>
<td></td>
<td></td>
<td>o Game clock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>o Winning conditions</td>
</tr>
<tr>
<td></td>
<td></td>
<td>o Aim &amp; Collision</td>
</tr>
</tbody>
</table>
Week 4.5 | 29th – 31st
---|---
• Game flow implementation
  o Character select
  o Map select
• Game mechanics
  o Default & Custom Gestures
  o Aim & Collision
• Process report writing

### November

<table>
<thead>
<tr>
<th>Week</th>
<th>Dates</th>
<th>Tasks</th>
</tr>
</thead>
</table>
| Week 1 | 1st – 4th | • Game mechanics
  • Game object modeling for haptic feedback device onto player model |
| Week 2 | 5th – 11th | • Game mechanics
  • Game object modeling for haptic feedback device onto player model |
| Week 3 | 12th – 18th | • Prototype testing & refinement (single player use case)
  • Haptic suit prototype |
| Week 4 | 19th – 25th | • Prototype testing & refinement (single player use case)
  • Haptic suit prototype |
| Week 4.5 | 26th – 30th | • Networking research
  • Haptic suit prototype
  • Process report writing |

### December

<table>
<thead>
<tr>
<th>Tasks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Presentation preparation over holidays</td>
</tr>
</tbody>
</table>

### January

<table>
<thead>
<tr>
<th>Week</th>
<th>Dates</th>
<th>Tasks</th>
</tr>
</thead>
</table>
| Week 1 | 1st – 6th | • Networking
  • Modeling
  • Animations & Audio |
| Week 2 | 7th – 13th | • Interim Presentation
  • Modeling
  • Animations & Audio
  • Report writing |
| Week 3 | 14th – 20th | • Networking
  • Modeling |
| Week 4 | 21st – 27th | • Interim Report |
| Week 4.5 | 28th – 31st | • Process report writing |

### February

<table>
<thead>
<tr>
<th>Week</th>
<th>Dates</th>
<th>Tasks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Week 1</td>
<td>1st – 3rd</td>
<td>• Two-player game mechanics testing</td>
</tr>
<tr>
<td>Week 2</td>
<td>4th – 10th</td>
<td>• Determine project process, determine whether AI component is feasible for single player experience</td>
</tr>
</tbody>
</table>
7. Conclusions

Virtual reality gaming is trending. However, the majority of VR games only engage the visual experience, and lesser-known VR supplemental technologies such as haptic feedback devices are not well supported, but there are many companies and startups that are trying to bring the spotlight back to the tactile perception. Therefore, we believe that now is a good time to develop a VR game that gives users a glimpse of what gaming will become. The second motivation behind this project, other than providing valuable user feedback, is the lack of multiplayer VR demos which feature haptic feedback support that are publically available.
8. References


