NATURAL HUMAN-COMPUTER INTERACTION APPLIED ON DESKTOP APPLICATIONS USING MULTIMODAL INPUT DEVICES

by

ANASTASIOS VLASOPOULOS

B, TEI of Crete, 2014

A THESIS

submitted in partial fulfillment of the requirements for the degree

MASTER OF SCIENCE

DEPARTMENT OF INFORMATICS ENGINEERING

SCHOOL OF APPLIED TECHNOLOGY

TECHNOLOGICAL EDUCATIONAL INSTITUTE OF CRETE

2016

Approved by:

Major Professor
Nikolaos Vidakis

Abstract

Human-Computer Interaction as a term that exists from the early 80s. It is an evolutional term which has its root from the term Man-Machine Interaction. Man-Machine Interaction as an idea came out of studies in the 1940s. Researchers from that period tried to understand how human interacts with any kind of machine and they wanted to realize how this interaction could be better and more efficient from the user experience perspective.

A lot of years have passed since the user was interacting with computers through difficult and not user-friendly interfaces like a command console for example. Today HCI is accomplished by as simple and user friendly interfaces as possible. User is in the center of this process and combination of knowledge in computer science, behavioral sciences, design, media studies, and several other fields is considered in order to make HCI a better user experience. If we take a look at today's HCI we can say that the way that human interacts with computer has become more natural. Computer is treated like a human and the means of interaction are more physical (speech, gestures or touch).

This thesis will be a proof of concept to claim that it is actually feasible to create an interface through which the user can interact with computer by a more natural and multimodal way. The implementation part will be a game that the user will interact with though natural means. This game will be an open source project which will be adapted in the terms of Natural HCI and multimodality. The way in which this will be accomplished is through the use of multimodal input devices. The first MID that will be used is the Microsoft Kinect and the second is the Leap. These two devices will be the main controllers of the game. Of course each of these devices will be analyzed how they work and it will be a guide of how they can be configured for further usage.

Finally the purpose of this thesis is to demonstrate how HCI is changed through years and how is being experienced today through an open source game. Machines plays a key role in human's life so it is very important for all of us to understand how humans interact with them in order to make our lives more easy and delectable. This thesis is a serious effort to accomplish this goal.

Table of Contents

Chapter 1 - Introduction

Evolution of technology over the last years has changed the relationship between man and computer in a way that human mind could not have been imagined a few decades ago. Multimodality and naturalness has become a common way of human computer interacting. Through a philosophical way of thinking human tend to treat computers like human if we analyze how the communication of these two entities has evolved the last few years.

A new concept concerning HCI is multimodality (multimodal interaction). According Bellik and Teil [1] is "a concrete form of a particular communication mode". Mode is defined as the five human senses which constitute the receiving information and the multifarious ways of human expression such as gesture, speech, facial expressions and others which constitute the product information.

A Natural User Interface (NUI) [2],[3] is a multimodal interface which gives the user the ability to interact with the computer with a natural way. In order to achieve the naturalness of the interaction, the NUI should be invisible after a successful learning. This transparency gives user the illusion of a natural interaction without a mean. So the naturalness of the interaction is being achieved. By this way the feeling of the user is that he interacts with the machine with the same manner as he interacts with humans. It have to be mentioned that in order to accomplish this goal, the right technology should be used and this technology should be easy learned by an average user.

Naturalness of interaction implies the usage of the right methods like body traction, speech recognition or eye tracking. This fact means that it should be used the right hardware devices like digital sensors for example. These devices offer high accuracy and efficiency but they are quite expensive. Additionally their specialization and their control difficulty has the effect of the hard accessibility to the users.

Of course the evolution of the technology is huge and rapid the last few decades. Companies like Microsoft and Assus have developed and released some affordable multimodal input devices that are small and easy to setup. Some of them are **Kinect** by Microsoft, the **Xtion Pro LIVE** by Assus, **The Leap** by Leap Motion or the **Google Glass** by Google. The above devices are small compared to the old one, but they demand specialized software programming in order to work properly and efficiently.

Summary

The evolution of the technology drives educational methods into new paths. A part of those methods are the games, especially when referring to preschool and young children. Education through entertainment started to be popular during the ‘90s and in general it includes gaming as a learning method. These kind of games are called educational or serious games.

Serious games are digital games that are used not only for entertainment. There main reason of existence is the learning through several experiences that users do not have the ability to live in the real world. They have a positive impact on the players’ skill development. With these games young people can be helped to improve their skills in reading or mathematics.

Focusing on the current thesis, its main purpose is to present a system that helps to achieve the natural user interaction between the user and the machine. The concept is that the user will be able to play any html serious game just with the movement of its finger, without the usage of the classical input devices like mouse or keyboard. The Leap by Leap Motion is used for this purpose as a sensor/input device for the achievement of natural user interaction.

Research questions

This section demonstrates the basic research questions that are being cited in this thesis.

i. How HCI affects the software applications?
ii. Is the Leap a suitable choice as a NUI device for the purpose of an application with educational content?
iii. Are there any limits that should be exceeded?

Structure

In this sub-section is analyzed the structure of the current thesis. Initially, the research question is sited. Afterwards the User Interface history is reviewed in order the reader to have an overview about how things in UI were evolved until today. Next follows a state of the art regarding Natural User Interfaces where the latest technologies and practices are demonstrated. Subsequently are cited the analysis and architectural design where the UML diagrams are shown. The next thesis part is about implementation technologies and environment and as next comes the environment configuration and integration instructions. Finally, the thesis' conclusion and feature work are cited.

Chapter 2 - User Interfaces History Review

A user interface (UI) from the perspective of human-computer interaction (HCI) is the intermediate between the human and the system. It allows the human to give the appropriate commands to the system. The model that UI is built on is request and response where human makes a requests waiting for the result feedback from the UI and vice versa.

Looking back on the evolution of the UI over the last decades it could be observed a significant development year by year. This evolution is explained in the next lines in order to understand the different requirements of UI over the years but also to realize how the human need for simplicity affects this technological “journey”.

Computer Interfaces (CI)

Firstly created computers on the early fifties offered almost no interaction at all [4]. The only way of interaction was the direct manipulation of the hardware which weighed over 20 tones and occupied a whole room. The programmer had to connect wires and reposition switches in order to give a certain command to the computer. The output feedback was coming out of printers and cards. ENIAC (Electronic Numerical Integrator And Computer) was the first digital general-purpose computer made [5]. It weighed over 30 tones, occupied 167m^2 and its electricity consumption was 150 kW. The input was card readers and the output was card punches.

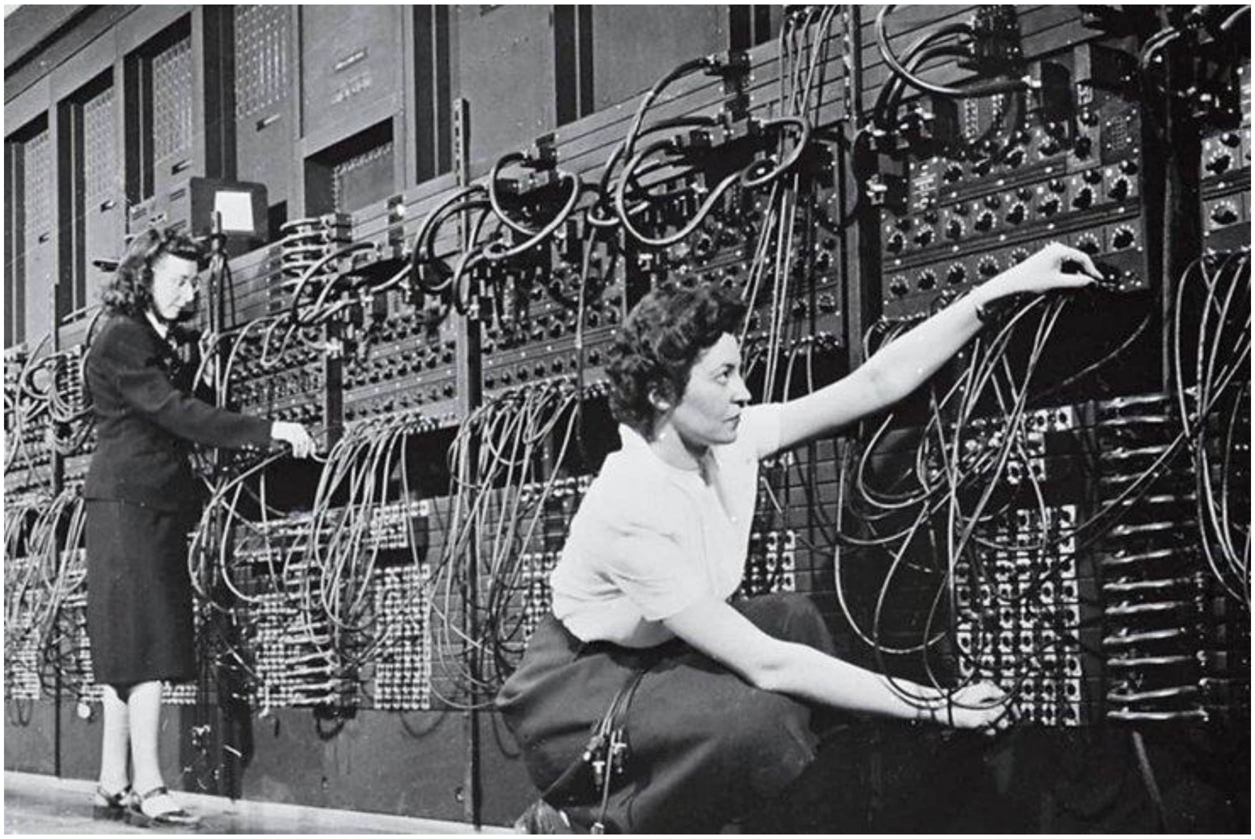
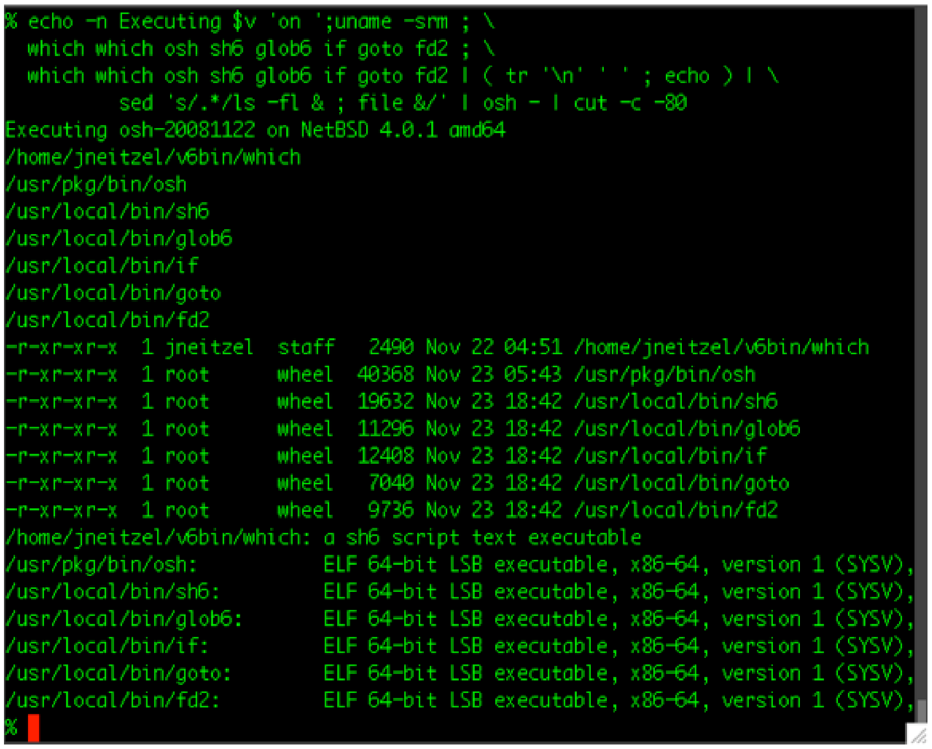


Figure 1: A programmer working with ENIAC, the first digital computer [6]

Command Line Interfaces (CLI)

In the early 60's video display terminal was introduced. That was the main factor for the development of the next generation of UI, the command language interpreter (CLI). Its alternative name is command-line user interface. CLI was the main way of interaction from 70's until late 80's, however some old school software engineers maintain this habit until now days despite the great evolution of UI on the last three decades. Regarding functionality, a command line shell is used in order to implement the interface. This shell is a software program that accepts text commands while translating them into operating system processes. Thompson shell, an implementation example of CLI, was the first Unix shell introduced in the first version of Unix in 1971. The image below shows an one of its instance.



```
% echo -n Executing $v 'on ';uname -srm ; \
  which which osh sh6 glob6 if goto fd2 ; \
  which which osh sh6 glob6 if goto fd2 | ( tr '\n' ' ' ; echo ) | \
          sed 's/.*/ls -fl & ; file &/' | osh - | cut -c -80
Executing osh-20081122 on NetBSD 4.0.1 amd64
/home/jneitzel/v6bin/which
/usr/pkg/bin/osh
/usr/local/bin/sh6
/usr/local/bin/glob6
/usr/local/bin/if
/usr/local/bin/goto
/usr/local/bin/fd2
-r-xr-xr-x  1 jneitzel  staff   2490 Nov 22 04:51 /home/jneitzel/v6bin/which
-r-xr-xr-x  1 root      wheel  40368 Nov 23 05:43 /usr/pkg/bin/osh
-r-xr-xr-x  1 root      wheel  19632 Nov 23 18:42 /usr/local/bin/sh6
-r-xr-xr-x  1 root      wheel  11296 Nov 23 18:42 /usr/local/bin/glob6
-r-xr-xr-x  1 root      wheel  12408 Nov 23 18:42 /usr/local/bin/if
-r-xr-xr-x  1 root      wheel   7040 Nov 23 18:42 /usr/local/bin/goto
-r-xr-xr-x  1 root      wheel   9736 Nov 23 18:42 /usr/local/bin/fd2
/home/jneitzel/v6bin/which: a sh6 script text executable
/usr/pkg/bin/osh:           ELF 64-bit LSB executable, x86-64, version 1 (SYSV),
/usr/local/bin/sh6:         ELF 64-bit LSB executable, x86-64, version 1 (SYSV),
/usr/local/bin/glob6:       ELF 64-bit LSB executable, x86-64, version 1 (SYSV),
/usr/local/bin/if:          ELF 64-bit LSB executable, x86-64, version 1 (SYSV),
/usr/local/bin/goto:        ELF 64-bit LSB executable, x86-64, version 1 (SYSV),
/usr/local/bin/fd2:         ELF 64-bit LSB executable, x86-64, version 1 (SYSV),
%
```

Figure 2: Thomson shell, the first Unix command line interface [7]

Graphical User Interfaces (GUI)

As the journey continues, we meet the third generation of user interfaces. The Graphical User Interfaces (GUI). Through GUI human can interact with an electronic system through visual elements and graphical artifacts. It was introduced in the 70's but they became popular in the early 80's. Xerox Alto computer, which was released in 1973, was one of the first electronic device that used a graphical user interfaces as the main user interface. However, Apple Macintosh was the first GUI-based electronic device which made a significant impact on broader public after the invention and establishment of the mouse as an alternative input device. Today's graphical user interfaces are quite sophisticated and they are designed based on particular principles which meet all the modern needs and standard of the human-computer interaction.

Figure 3: Xerox Alto, the first computer with GUI invented by researchers at the Stanford Research Institute [8]

Natural User Interfaces (NUI)

In the last decade Human Computer Interaction and User Interfaces have experienced a major evolution. Great efforts have lately been made in the hardware sector in order to gather information about human body movements, gestures, touch as well as speech with as more precision as possible. The result of this evolution is called Natural User Interfaces (NUI). NUI is based on a modern conceptual thinking that the interaction between human and computers should mimic the human-human interaction. Naturalness is the key artifact for this sector of HCI. The interaction is designed in such way that feels natural to the human, in which case intuition is achieved. Microsoft Kinect and The Leap are some of the latest sensors that support natural user interaction.

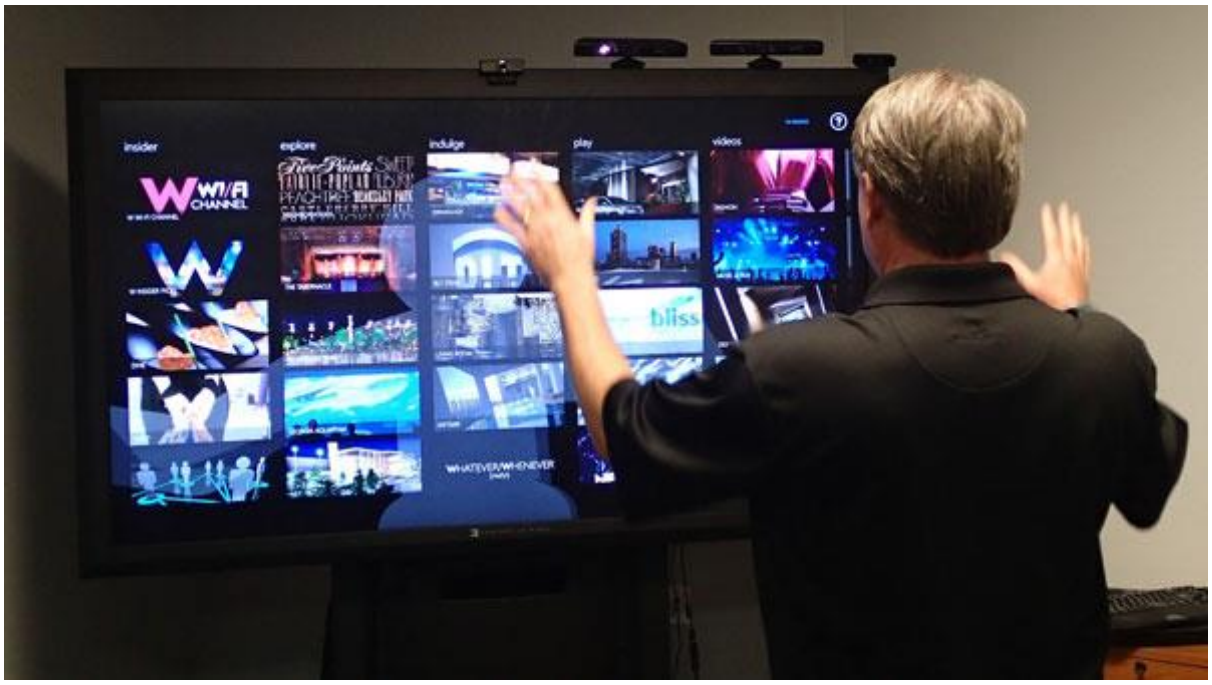


Figure 4: A modern NUI example. User interacts with the computer through his hands' gestures [9]

Chapter 3 - NUI Review - State Of The Art

Introduction

In this chapter, an overview about NUI sensors will be presented focusing on motion tracking area. As mentioned above NUI is in major progress in the last decade. The sensor evolution enables the theoretical concept of naturalness in human computer interaction to become gradually a modern reality. Nintendo Wii, PlayStation Move, Kinect and Leap motion are examples of the latest NUI sensors that became significant popular on Natural User Interfaces development. These devices will be further analyzed in the next sections of this chapter.

Nintendo Wii

Nintendo Wii released on November 19, 2006. It is a seventh-generation home video game console which recognizes body motion and rotation. The Wii introduced the Wii Remote controller which can be used as a handled pointing device and which can detect movement in three dimensions.

The Wii is Nintendo's smallest home console to date; it measures 44 mm wide, 157 mm tall and 215.4 mm deep in its vertical orientation, slightly larger than three DVD cases stacked together. The included stand measures 55.4 mm wide, 44 mm tall and 225.6 mm deep. The system weighs 1.2 kg, [10] making it the lightest of the three major seventh-generation consoles. The Wii may stand horizontally or vertically. After an upgrade to the device the new controller Wiimote Plus can identify exact orientation using a gyroscope.

In the tip of each Wii remote is an IR camera sensor manufactured by PixArt Imaging, shown in Figure 2. The camera chip features an integrated multi-object tracking (MOT) engine, which provides high-resolution, high-speed tracking of up to four simultaneous IR light sources. The camera sensor's exact specifications are unpublished, but it appears to provide location data with a resolution of 1,024 × 768 pixels, more than 4 bits of dot size or light intensity, a 100 Hz refresh rate, and a 45 degree horizontal field of view. The integrated hardware object tracking minimizes the data transmitted over the wireless connection and greatly simplifies the implementation of camera-based tracking applications [11].

Figure 5: A user playing a game on the Wii console [12]

PlayStation Move

PlayStation Move is another motion-sensing game controller platform. Actually it is a motion sensor which created by Sony and first released on 2 June 2009 for the PlayStation 3 (PS3) video console. It is based on a handheld wand. Inside that wand there are inertial sensors which main role is to detect the human’s body motion. Subsequently, the position of the wand is tracked be a webcam called PlayStation Eye.

PlayStation Move motion controller is the primary element of this device. As mentioned above it is a wand which allow human to interact with the gaming console through his motion and gestures. The PlayStation Move motion controller features an orb at the head which can glow in any of a full range of colors using RGB light-emitting diodes (LEDs)[13]. Based on the colors in the user environment captured by the camera, the system dynamically selects an orb color that can be distinguished from the rest of the scene. The colored light serves as an active marker, the position of which can be tracked along the image plane by the camera.[14] The uniform spherical shape and known size of the light also allows the system to simply determine the controller's distance from the camera through the light's image size, thus enabling the controller's position to be tracked in three dimensions [15] with high precision and accuracy.

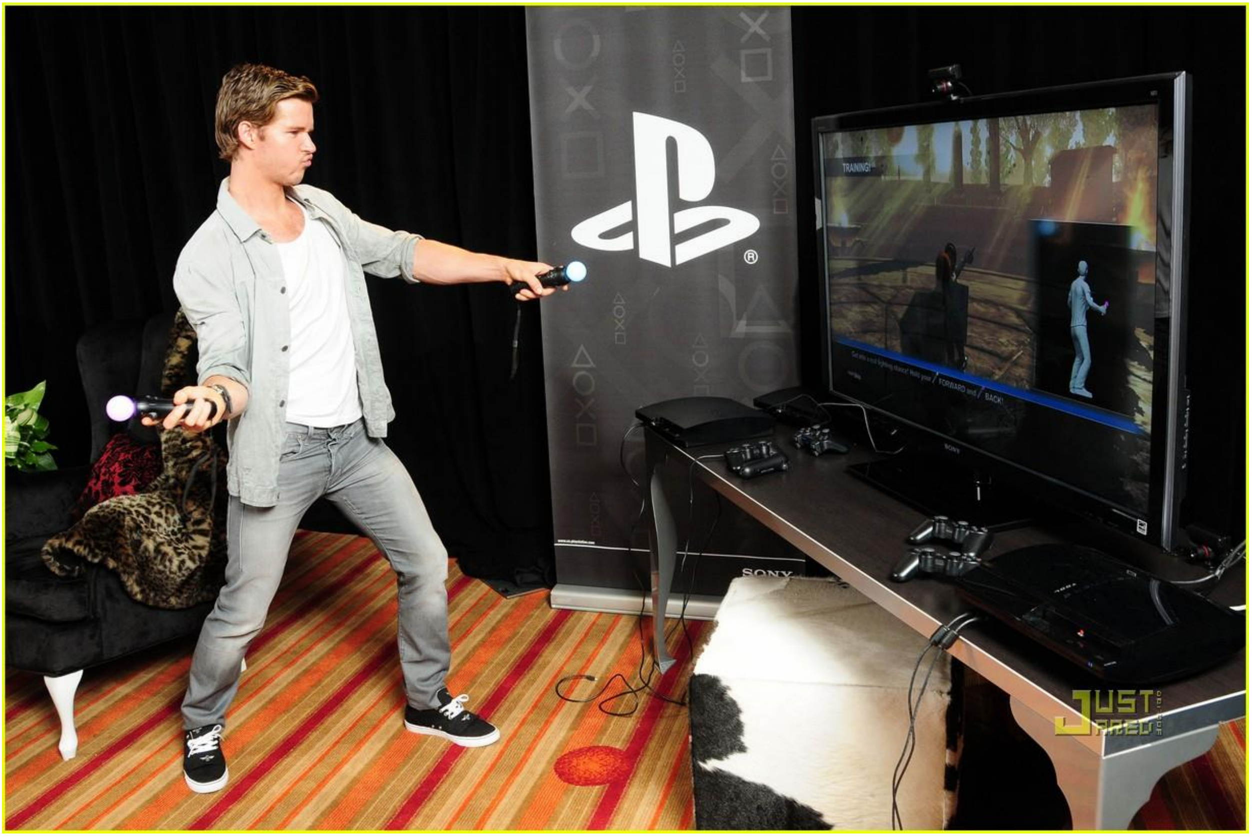


Figure 6: User playing a video game using PlayStation Move [16]

Kinect

Kinect is a hardware device developed by Microsoft. It was released in 2010 for the Xbox 360. For the first time in the history of game controllers user could interact with the system without touching anything. There is no user physical touch with any hardware interpreter in order to give an input. That was a breakthrough not only for the gaming community but also for the scientific field of Human Computer Interaction.

The Kinect device uses a 3D depth sensor, an RGB camera and a multi-array microphone to receive input from the user [17].

The depth sensor consists of an infrared projector and a monochrome CMOS sensor. In order for the Kinect to understand what exists in the scene and at which distance it needs a depth image of the scene. The IR projector emits infrared light on the scene. This light bounces off the surfaces of the objects in the scene and creates a pattern. The brighter the light that bounces off a surface, the closer that surface is to the Kinect. The darker it is, the further away it is from the device [18]. Using software that runs in the device, the Kinect can also predict what parts of the light pattern mentioned above belong to a person and therefore track the person's skeleton.

The Kinect tracks 20 joints on a skeleton. That information, the skeleton data, is made available to the application which can then compare the position of one or more joints to estimate the pose of the person tracked and whether or not a gesture is being performed. As a result the application can respond to different gestures.

Kinect Fusion provides 3D object scanning and model creation using a Kinect for Windows sensor. The user can paint a scene with the Kinect camera and simultaneously see, and interact with, a detailed 3D model of the scene. Although the Kinect Fusion can be run at interactive rates on supported GPUs, it also can run at non-interactive rates on a variety of hardware but allowing larger volume reconstructions.

Kinect Fusion in action, taking the depth image from the Kinect camera with lots of missing data and within a few seconds producing a realistic smooth 3D reconstruction of a static scene by moving the Kinect sensor around. From this, a point cloud or a 3D mesh can be produced. [19]

Light Coding Technology ™ [20] allows the Kinect device to create 3D depth maps of a scene also known as triangulation process in real time [21]. The laser source emits a single beam which is split into multiple beams from a diffraction grating, to create a stable pattern spots which are projected onto the surfaces and a CMOS image sensor (Complementary Metal Oxide Semiconductor) accepts the reflected rays. The PS1080 SoC chip (System on a Chip) made by PrimeSense that is contained inside the Kinect's system, controls the structure of light spots and processes the data from the CMOS sensor generating depth data in real time [22]. The maximum resolution of the depth image generated by the PS1080 is 640x480, with a frequency of 30 frames per second. At 2 meters distance from the sensor there is an accuracy of 3 mm in height and width and of 1 cm in depth. The range of proper operation is from 0.8 to 3.5 meters.

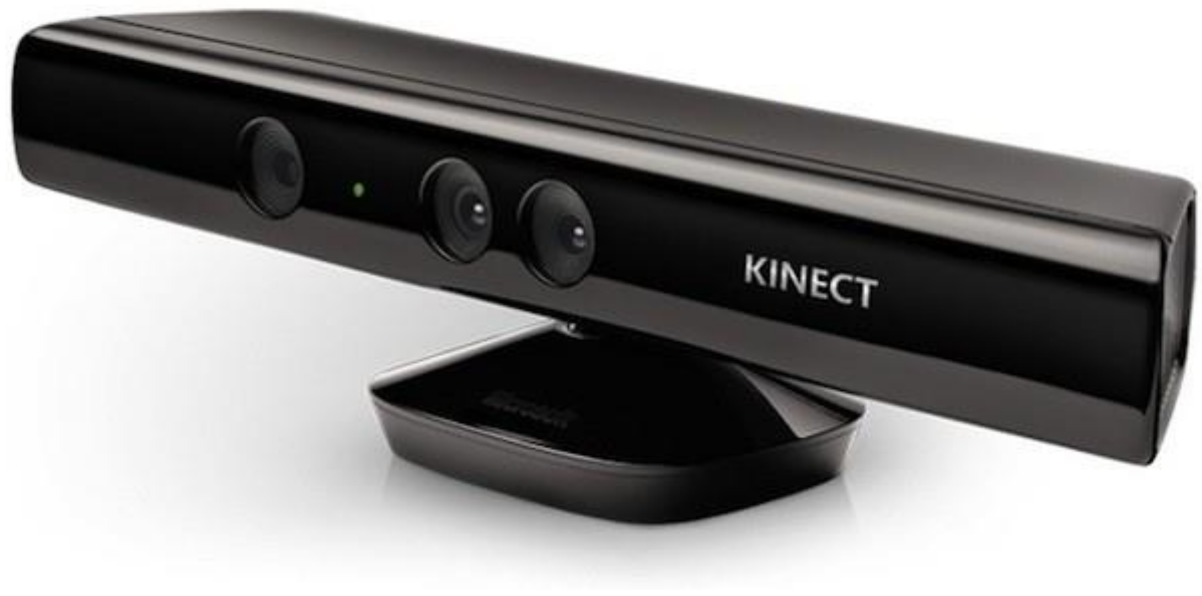


Figure 7: Microsoft Kinect device [23]

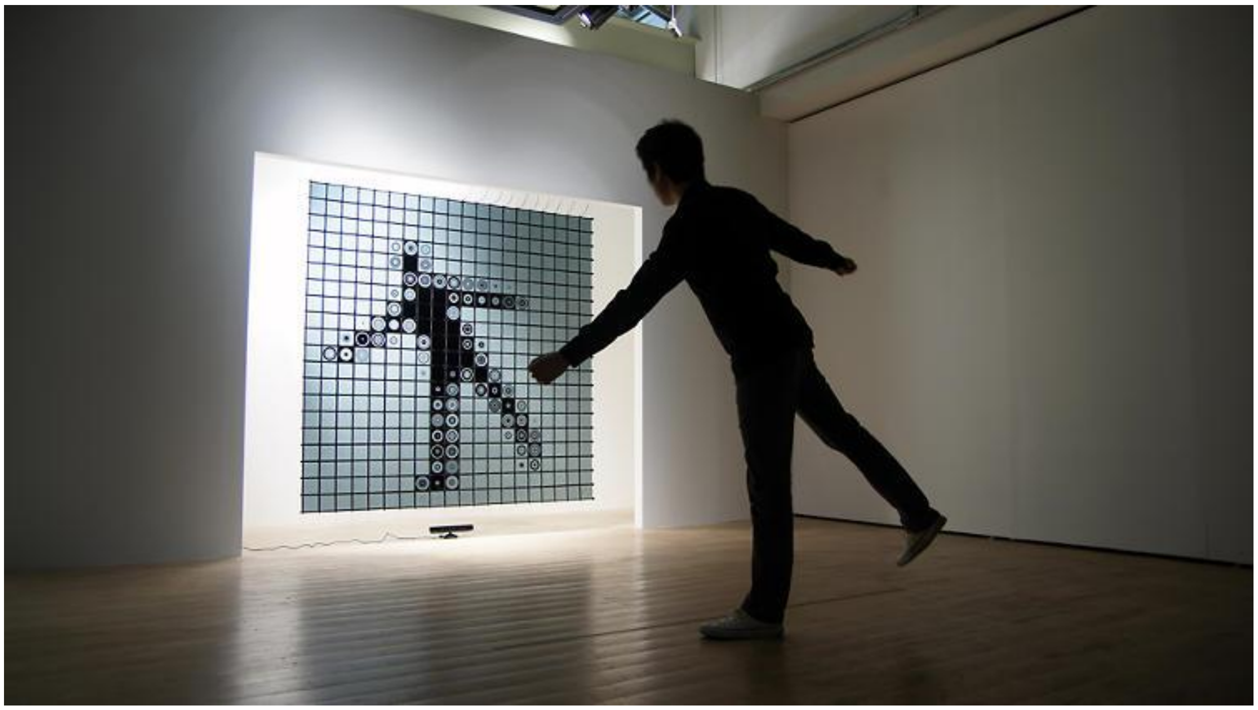


Figure 8: The user is interacting with a system through Microsoft Kinect sensor [24]

Leap Motion

The Leap Motion is a company that develops advanced motion detection technology to human-computer interaction. Originally inspired by the frustration around the 3D modeling, using the mouse and keyboard, they wanted to claim that accomplishing something in virtual reality, should be as easy as in real life. It supports hand and finger tracking without any physical contact by the user.

In 2008 was developed for first time the technology for Leap Motion. However on 21 May 2012 was publicly announced its first product and after that the company distributed 12000 units to some developers that were interested in creating software for the device. In 2014 the sensor was massively produced for the broader public.

The Leap motion controller is a small USB device. Its appropriate position is upward facing in a regular desk. It offers also the possibility to be mounted on to a virtual reality headset. Through the two monochromatic IR cameras and three infrared LEDs the device observes an area distance of 1 meter.

In contrast with Microsoft Kinect which is suitable for an entire body movement tracking, the Leap has a quite smaller area of observation and higher resolution. Consequently this device is better suitable for higher precision at finger tracking. It can be used for website navigation finger gestures, drawing as well as a way for accurate manipulation of complex 3D data visualizations. Additionally The Leap Motion controller has been used for by researchers and surgeons for medical software solutions due to the finger tracking precision that it offers.

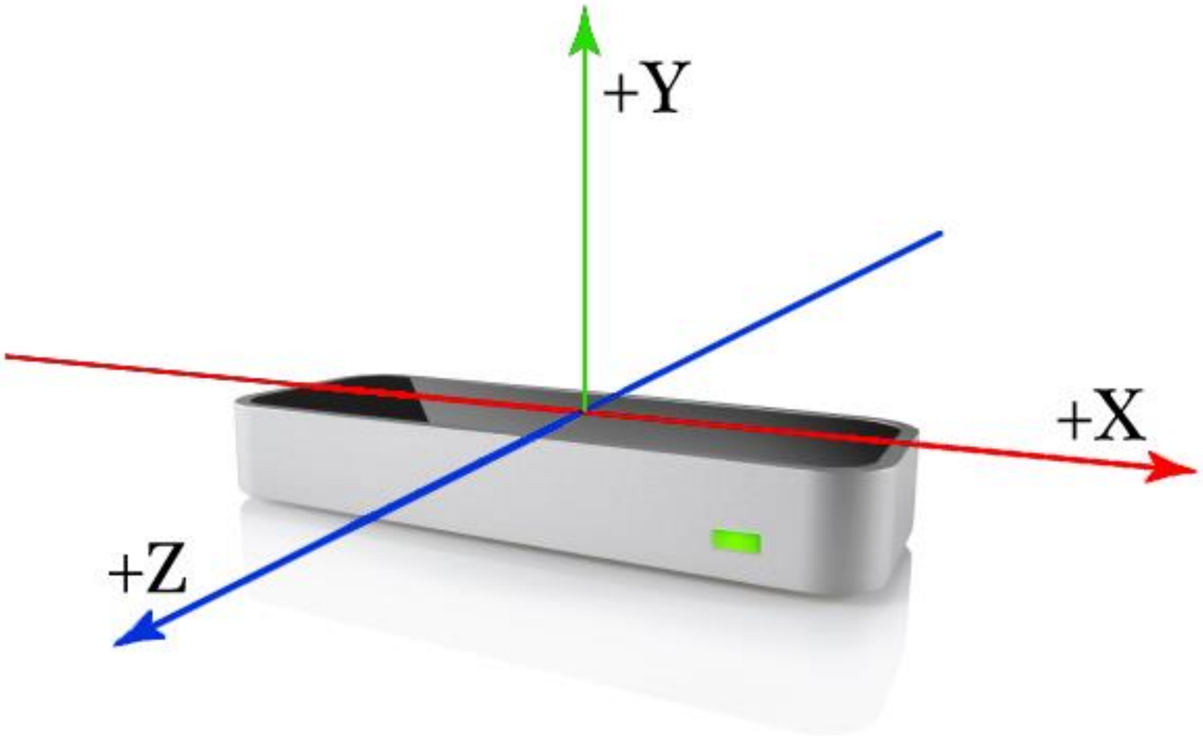


Figure 9: The three dimensions which are captured by The Leap [25]

Figure 10: User's interaction with the computer through the finger motion tracking that Leap offers [26]

Regarding the software, the Leap Motion Controller does not generate a depth map like Microsoft Kinect. It applies heavy and advanced algorithms to the raw sensor data in order to extract the accurate position of the fingers and hands. The software that processes the images is called The Leap Motion Service. This software analyze the environment and through image analyzation it reconstructs a 3D representation of what the device sees.[27]

Next, the tracking layer matches the data to extract tracking information such as fingers and tools. Our tracking algorithms interpret the 3D data and infer the positions of occluded objects. Filtering techniques are applied to ensure smooth temporal coherence of the data. The Leap Motion Service then feeds the results – expressed as a series of frames, or snapshots, containing all of the tracking data – into a transport protocol.

Through this protocol, the service communicates with the Leap Motion Control Panel, as well as native and web client libraries, through a local socket connection (TCP for native, WebSocket for web). The client library organizes the data into an object-oriented API structure, manages frame history, and provides helper functions and classes.

From there, the application logic ties into the Leap Motion input, allowing a motion-controlled interactive experience. Next week, we'll take a closer look at our SDK and getting started with our API. [27].

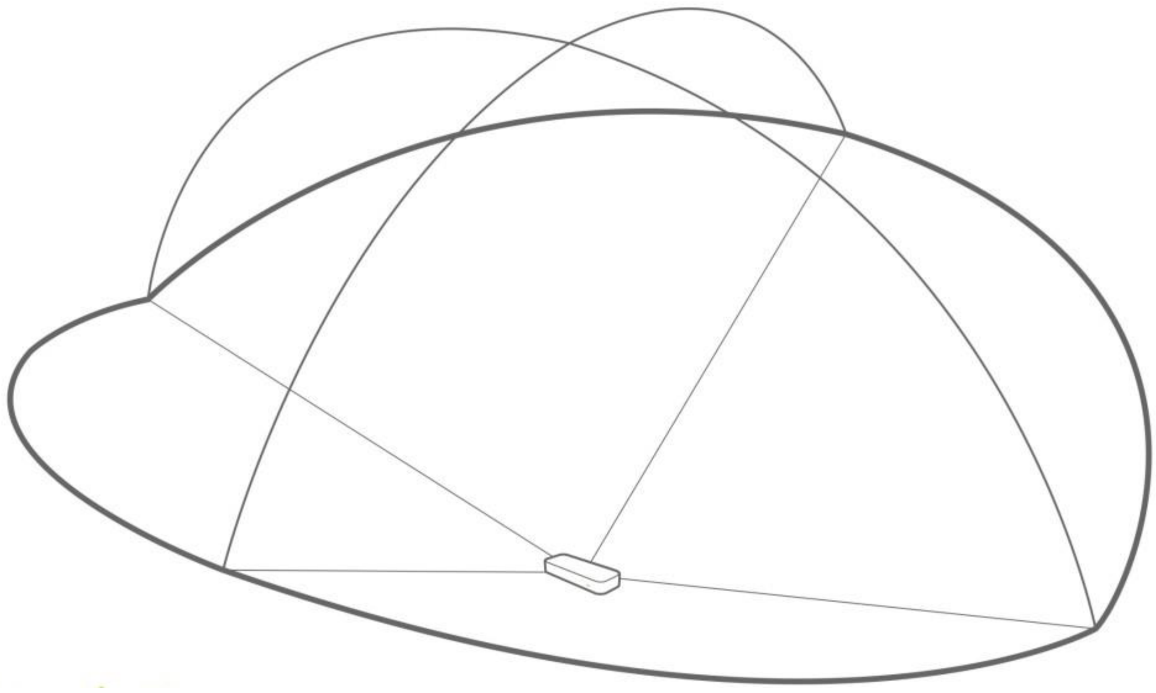


Figure 11: Interaction area of the controller [28]

The above figure demonstrates the interaction area of the Leap Motion controller. Initially this area covered 60 cm above the controller by 60 cm on each side (150° angle). Although with the Orion beta software the area limits have been expanded to 80 cm. This range is limited by LED light propagation through space, since it becomes much harder to infer your hand's position in 3D beyond a certain distance. LED light intensity is ultimately limited by the maximum current that can be drawn over the USB connection. [27]

Xtion Pro LIVE by Asus

In 2011 Asus released its first motion tracking sensor device, the Asus Wavi Xtion. This device was targeting the community of developers but it did not delivered the expected results and sales. After a few months Asus ceased its production before released an update that combined both a depth sensor, an RGB camera and microphones. Asus Wavi Xtion used only one USB port without the need of external power plugin which was the main advantage over the Microsoft Kinect [29].

Comparing to the Microsoft Kinect the advantages of Xtion Pro Live that it is more compact and lighter. It also has lower interference with another ASUS Xtion or PrimeSense device in Dual depth sensor configuration and the RGB image quality is quite higher than Kinect's. Regarding the negatives, it is a less popular device and it has lower driver quality. Additionally, it is not compatible with some USB controllers especially USB 3.0 compared to the Kinect.

Figure 12: Xtion Pro LIVE sensor by Asus [30]

MYO by Thalmic

In 2014 Thalmic, a small and unknown startup started a discussion about its new future released product, MYO. This sensor is a gesture recognition device which is worn on the forearm. The Myo enables the user to control technology wirelessly using various hand motions. Electromyographic sensors are used that are tracing the electrical activity in the forearm muscles. This technology is combined with gyroscope, accelerometer and magnetometer to recognize gestures. [31]

This was (and still is) a radical departure from the way gesture commands are typically registered. Most gesture-control systems, like Microsoft’s Kinect, still detect movements with cameras, which can be thrown off by poor lighting conditions, distance, and simple obstructions. By drawing gesture information directly from the user’s arm muscles instead of a camera, Myo circumvents all these problems and also works with devices that don’t have a camera in the first place.

To read the muscle activity in your forearm, Myo wraps it in eight different blocks, each of which contains a medical-grade EMG sensor. The armband also uses a three-axis gyroscope, three-axis accelerometer, and three-axis magnetometer to sense motion in any direction.

Muscle activity and motion readings are handled by an onboard ARM Cortex M4 Processor, which communicates with your devices via Bluetooth. For devices that don’t have Bluetooth functionality built in, Myo also comes with a Bluetooth dongle that plugs into any USB port.

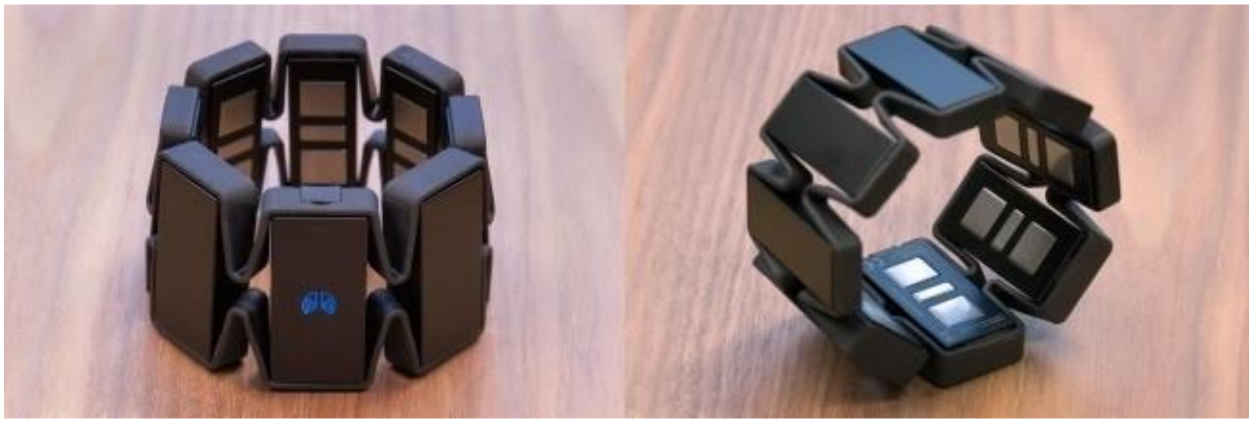


Figure 13: MYO a muscle tracking sensor device [32]

Oculus Rift by Oculus VR

On 28 March of 2016 the Oculus Rift was released. Almost two four years ago in 2012 Oculus started as a startup company with its main target to be the development of a VR device.

Rift is a virtual reality headset developed and manufactured by Oculus VR. Various pre-production models have been released which were referred to the developers’ community. However, because of the large number of people wanted to live the experience of this technology, Oculus was purchased broadly. The introductory price is 599$.

The Rift consists of an OLED display whit a 1080X1200 resolution per eye which has a 90 Hz refresh rate and a 110° field of view. The intergraded headphones provide a 3D audio effect, rotational and positional tracking. The positional tracking system, called "Constellation", is performed by a USB stationary infrared sensor that is picking up light that is emitted by IR LEDs that are integrated into the head-mounted display. The sensor normally sits on the user's desk. This creates 3D space, allowing for the user to use the Rift while sitting, standing, or walking around the same room.[33]

Initially the Rift was designed as a gaming device focused in gaming. However, it is used to view videos from inside a virtual cinema environment, giving the user the perception of viewing the content on a cinema sized screen.[34] Apart from that a number of social applications are being developed because of the Oculus’ belief that social applications will be quite popular in the virtual reality in the near future. Additionally, the Rift is experimentally used by a number of architecture firms for visualization and design purposes. Therefore this device could be applicable to the industrial and professional sector. Finally, Fox Sports began producing content for Oculus Rift device in 2016. As a conclusion it can be claimed that this device creates many expectations regarding the future evolution of Human Computer Interaction.

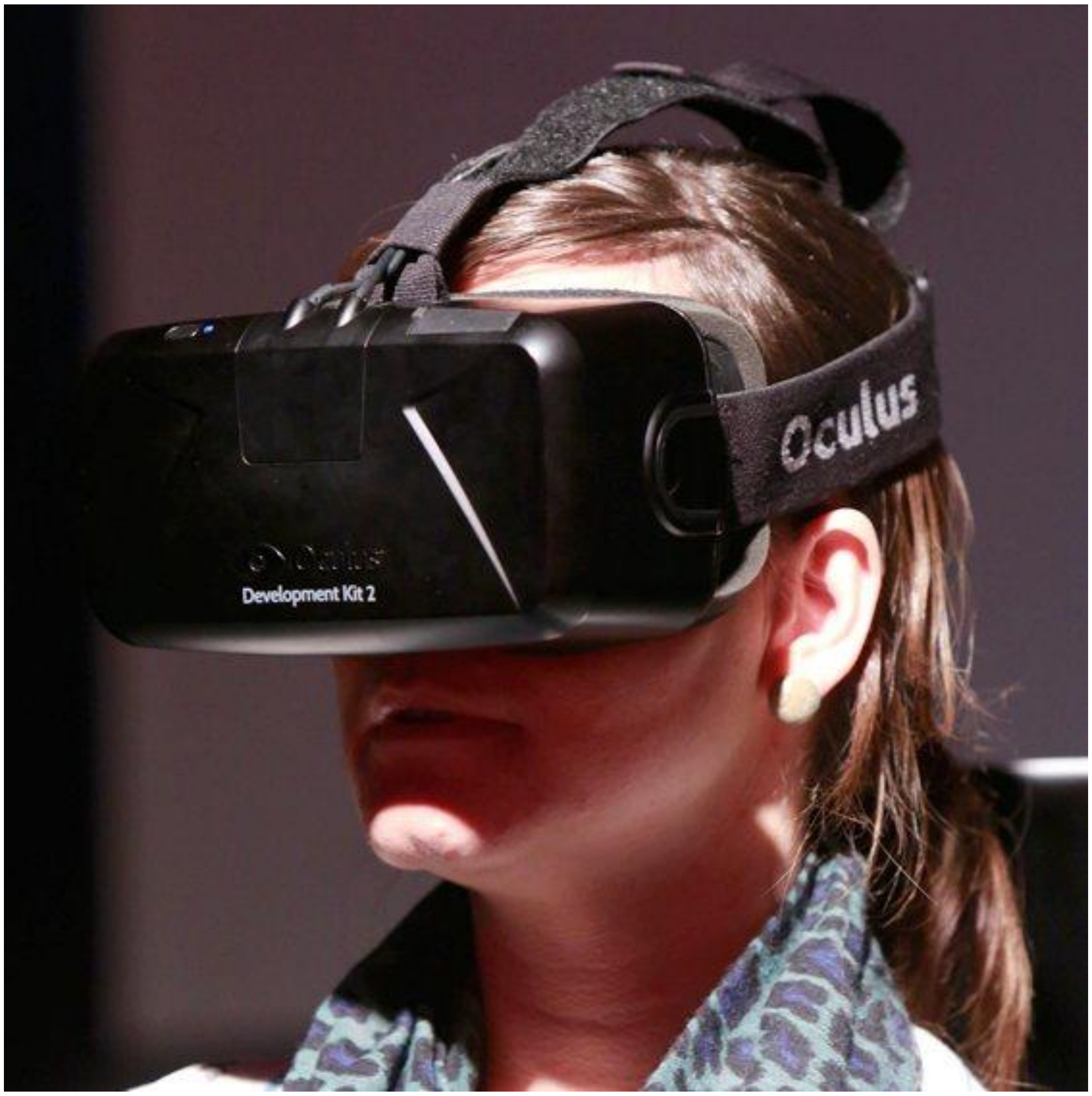


Figure 14: Oculus Rift device in action [35]

Google Glass by Google X

Google Glass is an optical head-mounted display designed in the shape of a pair of eyeglasses. It was developed by Google X with the mission of producing a ubiquitous computer. The first prototype was released on 15 April 2014 for 1500$. On 15 May 2014 the device became available to the public.

Google Glass is a concept that brings a hands free display technology to the users using a wireless date connection. Through this connection it bring information to the user from several Google sources.

Some of the features that offers is touchpad, camera and display. The touchpad is located on the side of Google Glass, allowing users to control the device by swiping through a timeline-like interface displayed on the screen. Sliding backward shows current events, such as weather, and sliding forward shows past events, such as phone calls, photos, circle updates, etc. The camera that includes has the ability to take photos and record 720p HD video. Finally, the display is made of a liquid crystal on silicon (LCoS) illuminated by LED technology. The panel reflects the light and alters it to S-polarization at active pixel sensor sites. The out-coupling beam splitter, which is a partially reflecting mirror, not a polarizing beam splitter, reflects the collimated light another 45° and into the wearer's eye.

Regarding the software, the applications that are included in the hardware are free and were built by third party developers. Additionally, Google Glass uses many of the existing Google applications like Google Now, Google Maps or Gmail. Plenty of firms and individual developers have built applications for this device like social network or facial recognition applications.

At this point it has to be mentioned that there has been an inconclusive debate about whether the use of this device brings in an intrusion of privacy. Additionally, there is controversy that Google Glass would cause security problems and violate privacy rights.[36] Organizations like the FTC Fair Information Practice work to uphold privacy rights through Fair Information Practice Principles (FIPPS), which are guidelines representing concepts that concern fair information practice in an electronic marketplace [37]. Some companies in the U.S. have posted anti-Google Glass signs in their establishments.

Other concerns have been raised regarding legality of the Glass in a number of countries, particularly in Russia, Ukraine, and other post-Soviet countries. In February 2013, a Google+ user noticed legal issues with Glass and posted in the Glass Explorers community about the issues, stating that the device may be illegal to use according to the current legislation in Russia and Ukraine, which prohibits use of spy gadgets that can record video, audio or take photographs in an inconspicuous manner.

Concerns were also raised in regard to the privacy and security of Glass users in the event that the device is stolen or lost, an issue that was raised by a US congressional committee. As part of its response to the committee, Google stated that a locking system for the device is in development. Google also reminded users that Glass can be remotely reset.[38] Police in various states have also warned Glass wearers to watch out for muggers and street robbers.[39]

TECHNICAL SPECIFICATIONS
Android 4.4
640×360 Himax HX7309 LCoS display
5-megapixel camera, capable of 720p video recording
Wi-Fi 802.11b/g
Bluetooth
16 GB storage (12 GB available)
Texas Instruments OMAP 4430 SoC 1.2Ghz Dual(ARMv7)
1 GiB RAM
3 axis gyroscope
3 axis accelerometer
3 axis magnetometer (compass)
Ambient light sensing and proximity sensor
Bone conduction audio transducer

The above table demonstrates the technical specifications of Google Glass. These specifications are for the developer Explorer unit version 2. In version 1 these may have slight differences.

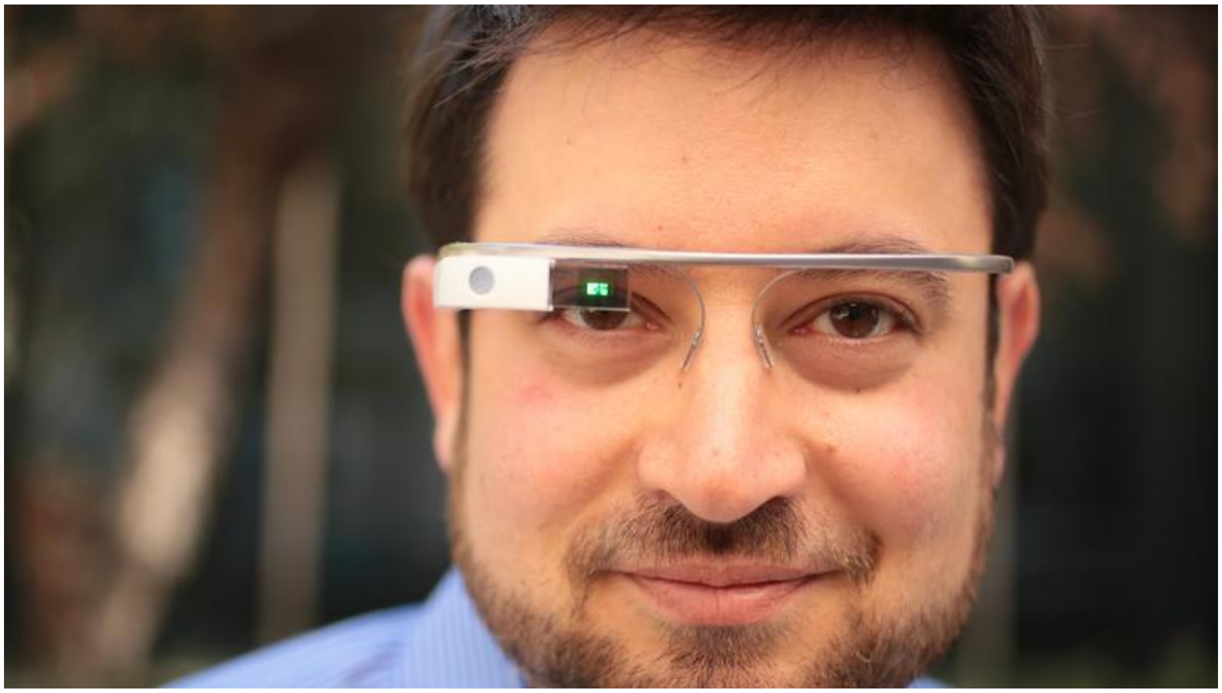


Figure 15: Google glass device worn by a user [40]

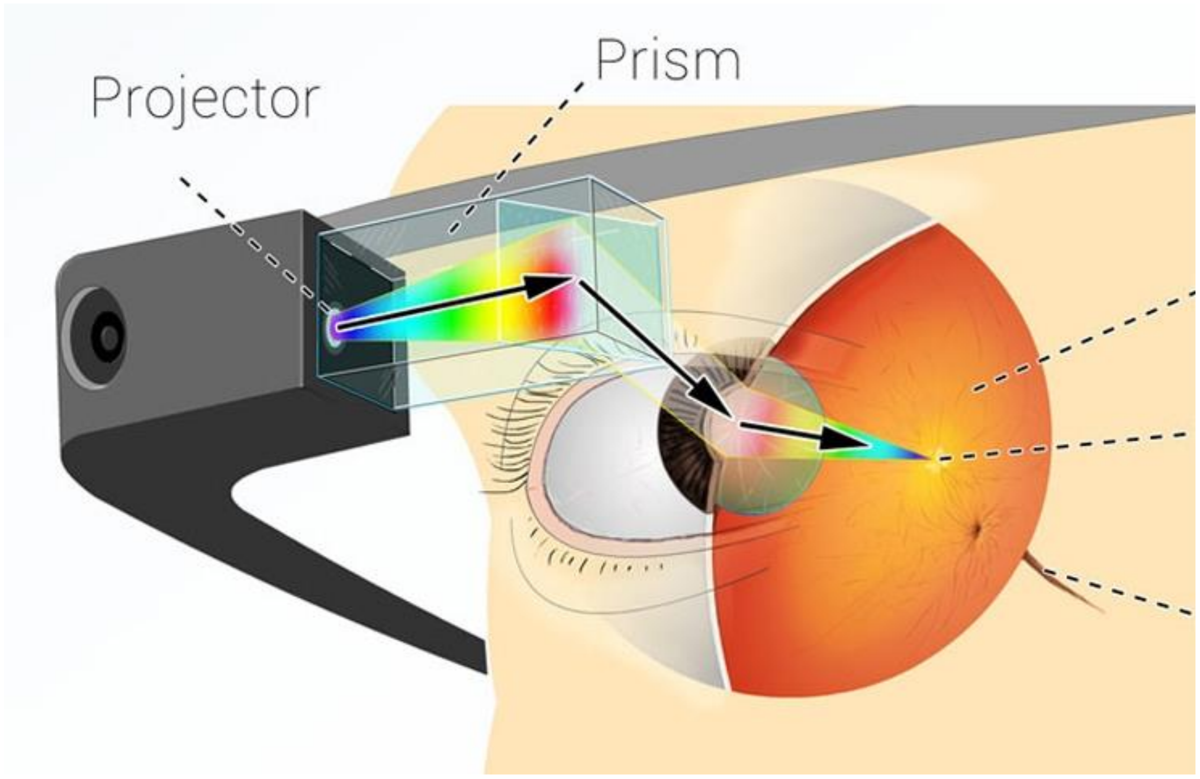


Figure 16: How Google Glass works [41]

Blended Learning

Over the years, pedagogical methods have evolved and consistently improved, compared to the educational systems of the past. A significant factor of this progress is unquestionably the increasing use of technology in teaching. Nowadays, most educators prefer to blend traditional teaching with interactive software, in order to achieve the maximum involvement of their students and to consolidate their learning.

“Blended learning” is a term which has yet to be universally defined. A sufficient description could be the following: “Blended learning is the process of using established teaching methods merged with Internet and multimedia material, with the participation of both the teacher and the students” [42].

Still, there are a few matters in question regarding the process of creating blended learning environments [43] which are the following.

Firstly, there is the issue of the importance of face to face interaction. Several learners have stated that they are more comfortable with the part of live communication in merged teaching methods, considering it more effective than the multimedia based part. Others are of the exact opposite opinion, which is that face to face instruction is actually not as required for learning, as it is for socialization. There are also those who believe that both live interaction and online or software material are of the same significance and it is the learner’s decision which of the two is more suitable for their educational needs.

Secondly, there is the question of whether the learners opt for combined learning exclusively because of the flexibility and accessibility of the method, without taking into account if they are choosing the appropriate type of blended teaching. Also, there are doubts regarding the ability of the learners to organize their own learning without the support of an educator.

Another matter, is the need for the instructors to dedicate a large amount of time to guide the learners and to equip them with the necessary skills in order to achieve their goals. In addition, the instructors need to continue being educated themselves to be able to meet the requirements of blended teaching.

There is also the argument that schools which have integrated technology into the curriculum are mostly addressed and beneficial to people in a comparatively favorable position in terms of economic or social circumstances. This, however, is refuted by the fact that blended teaching methods are quite affordable and easily accessible. The quick distribution of the multimedia material used in this type of instructing, is considered an advantage, but the universality of the system raises the need to modify the provided material, in order to make it more culturally appropriate for each audience.

Lastly, a continuous effort is being made to proceed to the new directions given by the novelties of technology while maintaining a low-priced production of educational material. The uninterrupted evolution of communication and information technology is making this effort rather strenuous for the developers of such teaching models.

Despite the difficulties faced when designing blended learning techniques, the advantages of combining face to face instructing with technology are many. Some of enormous importance [44] are:

1. The learner is intrigued by the procedure itself and as a result, the material being taught seems more interesting to them. This leads to the easier accomplishment of the educational goals that the teacher has set.

2. There is no limitation regarding the time or the place of the lesson. A student can attend remote classes being offered online, frequently having the opportunity to record and watch again the lesson.

3. The cost of the teaching process is greatly reduced. Every school unit that uses blended learning, has access to a variety of Internet and software material which would require a lot of time and effort to be independently produced, resulting in additional expenses.

4. In industrial applications of blended learning, it has been observed that the desired results have been achieved twice as fast as with the established instruction methods.

Regardless of how little official research has been done on blended learning, there are Universities that provide substantial information on the matter [44]. Stanford University is an institution with much experience in individually oriented programs for charismatic people. An issue which troubled the University was that a great percentage of their students (a bit less than half) would not finish the programs. They concluded that the fault was in the way the students were being taught, compared with the way they wished they would be taught. Thus, they decided to include live eLearning in their programs, leading to a spectacular rise in the completion percentage, which then reached 94%. The factors that were credited to affect the scholars in such a positive manner were: (a) the scheduled live interaction, which provided inspiration and incentive to finish the assignment without delay, and (b) the access to communication with numerous teachers and fellow students, which gave a chance to experience superior forms of instructing and learning.

The example of Stanford University proves that joining customized educational material with live eLearning interaction could facilitate both the participation in the program and the completion of it. These findings, can motivate other institutions and companies to invest in self-paced teaching systems combined with multimedia resources.

However, blended learning should not be conceived solely as a method of enriching the class with technology or making the learning process more accessible and engaging. Its main purpose is to modify and adjust the teaching and learning interaction in order to upgrade it. It assists and enables the growth of critical thinking, creativity and cognitive flexibility [45].

Chapter 4 - Implementation’s Technologies & Environment

This chapter provides the appropriate information regarding the technical tools and environment that have been used for the development of the current theses’ implementation. The following sections are focused on the programming language, version control system, build system and integrated development environment that have been utilized.

Programming Language

Orion is the latest SDK for the Leap Motion Controller, while it supports nine programming languages: Leap C, C++, C#, Unity, Objective-C, Java, Python, JavaScript and Unreal Engine. Regarding the thesis implementation, Java has been chosen as the most suitable programming language. The criteria for this choice are cited below.

Java is an excellent language for programming cross platform applications, since a Java application can run on every platform that has a Java Virtual Machine (JVM). By that means that the developer does not need to recompile the application before distributing it to the target system. Once that is done clients will be able to run without an issue. This is very important because the developer does not have to care about the different platform environments where the application will run on, like Linux or Windows for example. JVM offers an abstract layer between the software and the operating system, so the developer’s effort is spent only for the logic implementation of the software.

Additionally, because of the language popularity the last years have been developed a great number of tools that make the developer’s life more simple and sophisticated. Some of the belong to the IDE category like Netbeans, Eclipse or IntelliJ. Apart from that several tools for troubleshooting, profiling, monitoring and management exist and are maintained by Oracle for the efficient usage of the language.

Also, using Java is free. At this point it should be mentioned that JVM and JDK usage are subject to certain license terms and conditions. In order to bypass this issue the above technologies could be made from scratch without coping any of the original code or including the Java Class libraries. By this way any liability to Oracle could be avoided but it could run afoul of patents.

Apart from that, Java has a huge amount of documentation for every of its versions. It is also so detailed that even a developer with little experience could read and understand how to use every single Class, Interface or Enumeration. The documentation is supported and maintained by a large open community and it has many stages before being published.

At this point it is has to be admitted that there is a large pool of developers available, since Java language is very popular among programmers because of its usability and efficiency. By this means that it is very easy to find people with or without experience that would like to work on a project based on Java. This is very important because the community is large and the amount of people who could work on a project is great.

A technical advantage is that the memory is managed by the Garbage Collector, not the developer unlikely to C for example. This is a great advantage because the developer does not have to worry about the memory. This is handed by the JVM in a very efficient and sufficient way. Of course there is always a way for developer to instantly call the Garbage Collector manually but this is not a best practice at all.

Another technical advantage is that Java is a multithreaded language. This means that it could be developed applications that would exploit the latest powerful CPUs by the best way. The performance of the language rises and the its efficiency is very high.

Finally, this language is suitable not only for desktop but also for client-server model application as well as for web development. Java EE brings web development to the foreground and the developer is able to create secure and robust web applications.

Build System

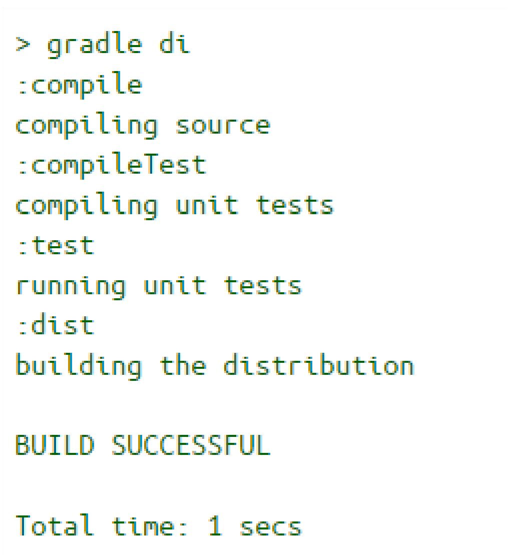
Gradle has been selected for the main build system of the thesis application. It is an open source build automation system. The concept that this system was built upon is the concept of Apache Ant and Apache Maven In general this is a modern build system tool which is suitable for Java, Groovy and Scala applications, but in the future more languages will be supported. It was designed for multi-project builds that can grow to a quite large project application. The benefits of Gradle are sited below.

Firstly, it can execute multiple tasks. The developer can execute multiple tasks in a single build by listing each of the tasks on the command-line. For example, the command gradle compile test will execute the compile and test tasks. Gradle will execute the tasks in the order that they are listed on the command-line, and will also execute the dependencies for each task. Each task is executed once only, regardless of how it came to be included in the build: whether it was specified on the command-line, or as a dependency of another task, or both.

Secondly, the developer can exclude tasks from being executed using the -x command line option and providing the name of the task to exclude.

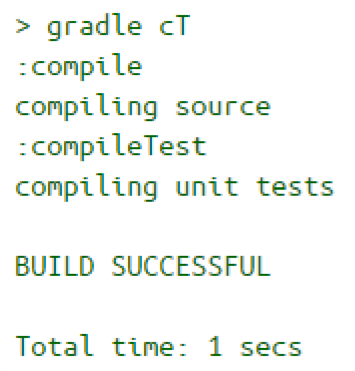
Additionally, is offers the ability to continue the build when a failure occurs. By default, Gradle will abort execution and fail the build as soon as any task fails. This allows the build to complete sooner, but hides other failures that would have occurred. In order to discover as many failures as possible in a single build execution, the developer can use the --continue option. When executed with --continue, Gradle will execute every task to be executed where all of the dependencies for that task completed without failure, instead of stopping as soon as the first failure is encountered. Each of the encountered failures will be reported at the end of the build. If a task fails, any subsequent tasks that were depending on it will not be executed, as it is not safe to do so. For example, tests will not run if there is a compilation failure in the code under test; because the test task will depend on the compilation task (either directly or indirectly).

When the developer specifies tasks on the command-line, he does not have to provide the full name of the task. He only needs to provide enough of the task name to uniquely identify the task. For example, in the sample build above, you can execute task dist by running gradle d. He can also abbreviate each word in a camel case task name. For example, he can execute task compileTest by running gradle compTest or even gradle cT.



```
> gradle di
:compile
compiling source
:compileTest
compiling unit tests
:test
running unit tests
:dist
building the distribution

BUILD SUCCESSFUL

Total time: 1 secs
```

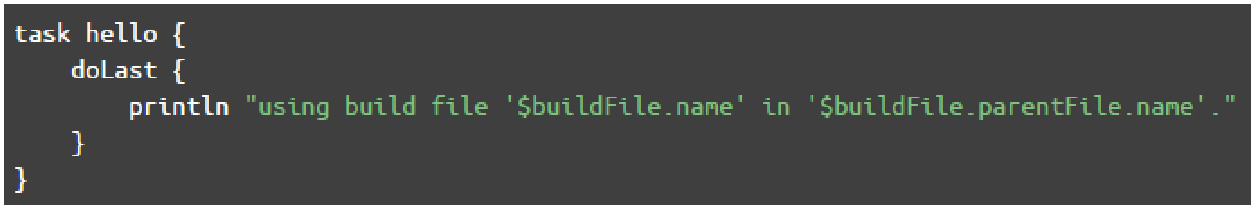
Figure 17: Example of abbreviated task name.



```
> gradle cT
:compile
compiling source
:compileTest
compiling unit tests

BUILD SUCCESSFUL

Total time: 1 secs
```

Figure 18: Example of abbreviated camel case task name.

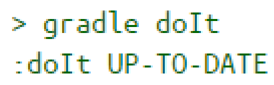
When the developer runs the gradle command, it looks for a build file in the current directory. He can use the -b option to select another build file. If he uses -b option then settings.gradle file is not used. The example below explains this issue.



```
task hello {
    doLast {
        println "using build file '$buildFile.name' in '$buildFile.parentFile.name'."
    }
}
```

Figure 19: Example of selecting the project using a build file.

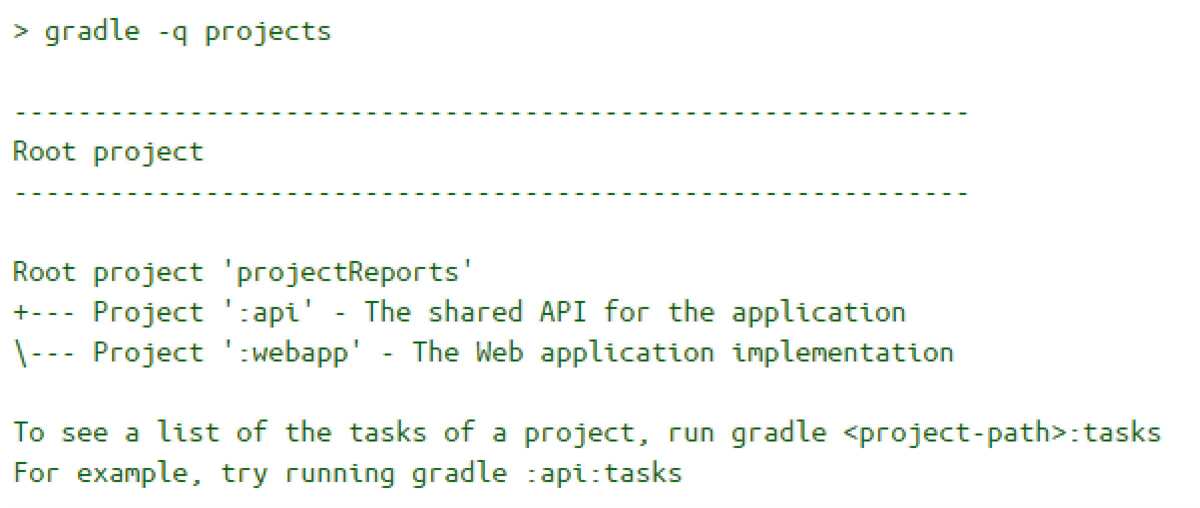
Many tasks, particularly those provided by Gradle itself, support incremental builds. Such tasks can determine whether they need to run or not based on whether their inputs or outputs have changed since the last time they ran. The developer can easily identify tasks that take part in incremental build when Gradle displays the text UP-TO-DATE next to their name during a build run. He may on occasion want to force Gradle to run all the tasks, ignoring any up-to-date checks. If that's the case, he could simply use the --rerun-tasks option. Below is demonstrated the output when running a task both without and with –rerun-tasks.



```
> gradle doIt
:doIt UP-TO-DATE
```

Figure 20: Example of forcing tasks to run.

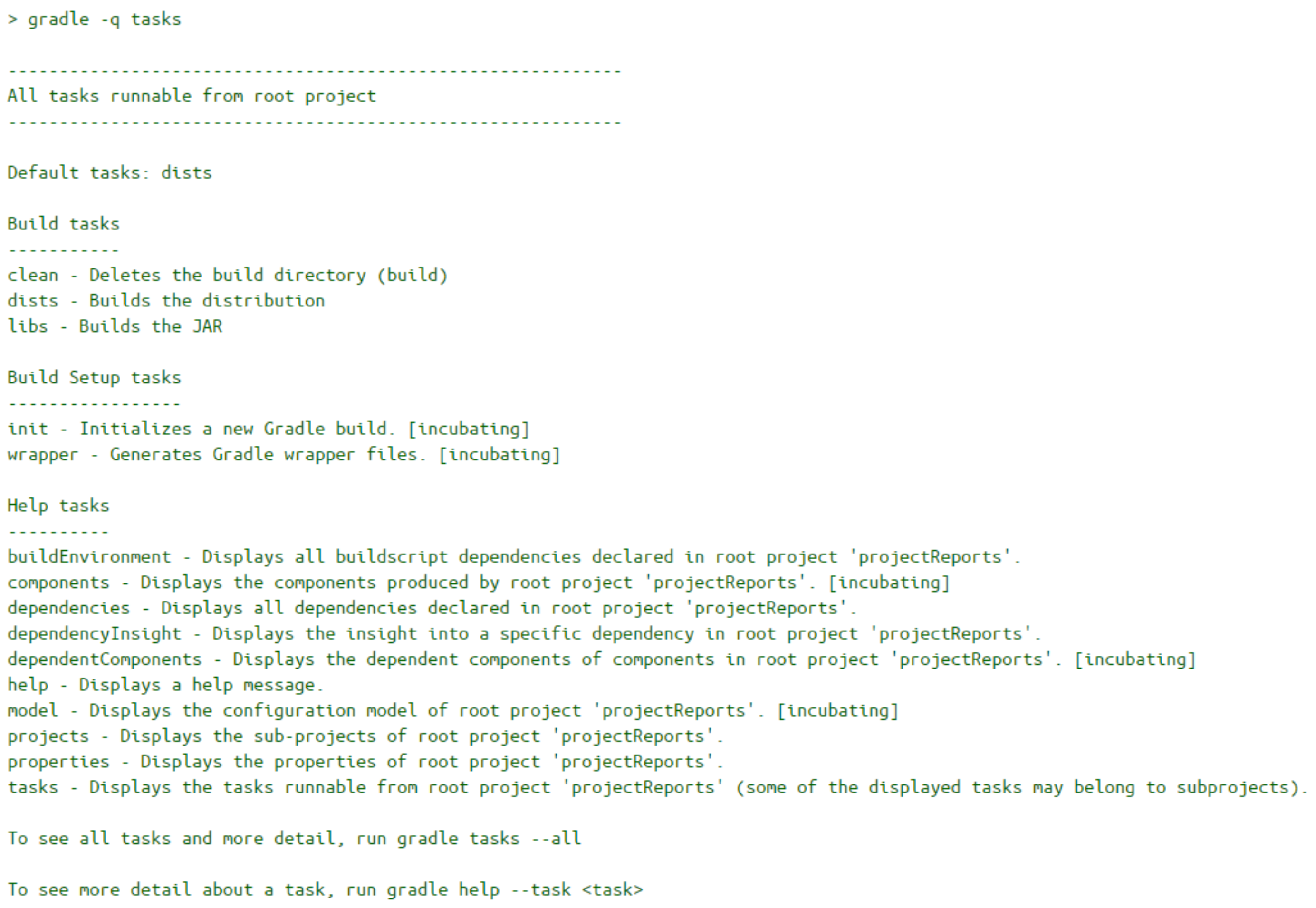
Additionally, Gradle provides several built-in tasks which show many details of a build. This can be useful for understanding the structure and dependencies of the build, and for debugging problems. In addition to the built-in tasks shown below, the developer can also use the project report plugin to add tasks to the project which will generate these reports.



```
> gradle -q projects

------------------------------------------------------------
Root project
------------------------------------------------------------

Root project 'projectReports'
+--- Project ':api' - The shared API for the application
\--- Project ':webapp' - The Web application implementation

To see a list of the tasks of a project, run gradle <project-path>:tasks
For example, try running gradle :api:tasks
```

Figure 21: Example of obtaining information about projects

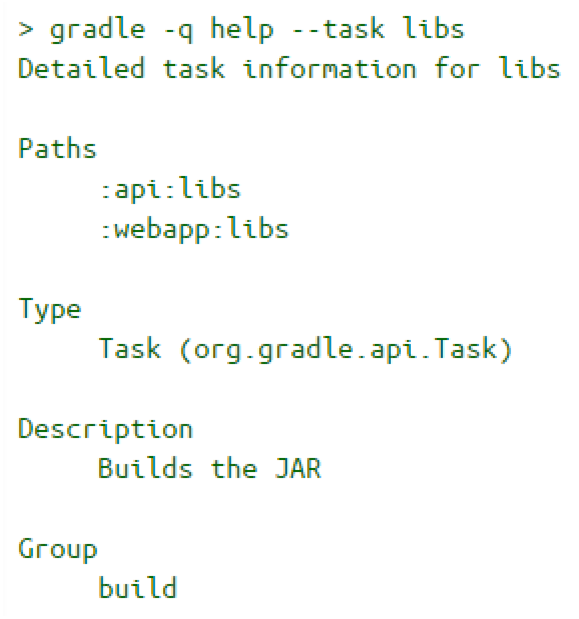
Running gradle tasks gives you a list of the main tasks of the selected project. This report shows the default tasks for the project, if any, and a description for each task. Below is an example of this report.



```
> gradle -q tasks

------------------------------------------------------------
All tasks runnable from root project
------------------------------------------------------------

Default tasks: dists

Build tasks
-----------
clean - Deletes the build directory (build)
dists - Builds the distribution
libs - Builds the JAR

Build Setup tasks
-----------------
init - Initializes a new Gradle build. [incubating]
wrapper - Generates Gradle wrapper files. [incubating]

Help tasks
----------
buildEnvironment - Displays all buildscript dependencies declared in root project 'projectReports'.
components - Displays the components produced by root project 'projectReports'. [incubating]
dependencies - Displays all dependencies declared in root project 'projectReports'.
dependencyInsight - Displays the insight into a specific dependency in root project 'projectReports'.
dependentComponents - Displays the dependent components of components in root project 'projectReports'. [incubating]
help - Displays a help message.
model - Displays the configuration model of root project 'projectReports'. [incubating]
projects - Displays the sub-projects of root project 'projectReports'.
properties - Displays the properties of root project 'projectReports'.
tasks - Displays the tasks runnable from root project 'projectReports' (some of the displayed tasks may belong to subprojects).

To see all tasks and more detail, run gradle tasks --all

To see more detail about a task, run gradle help --task <task>
```

Figure 22: Example of obtaining information about tasks. Output of galdle -q task

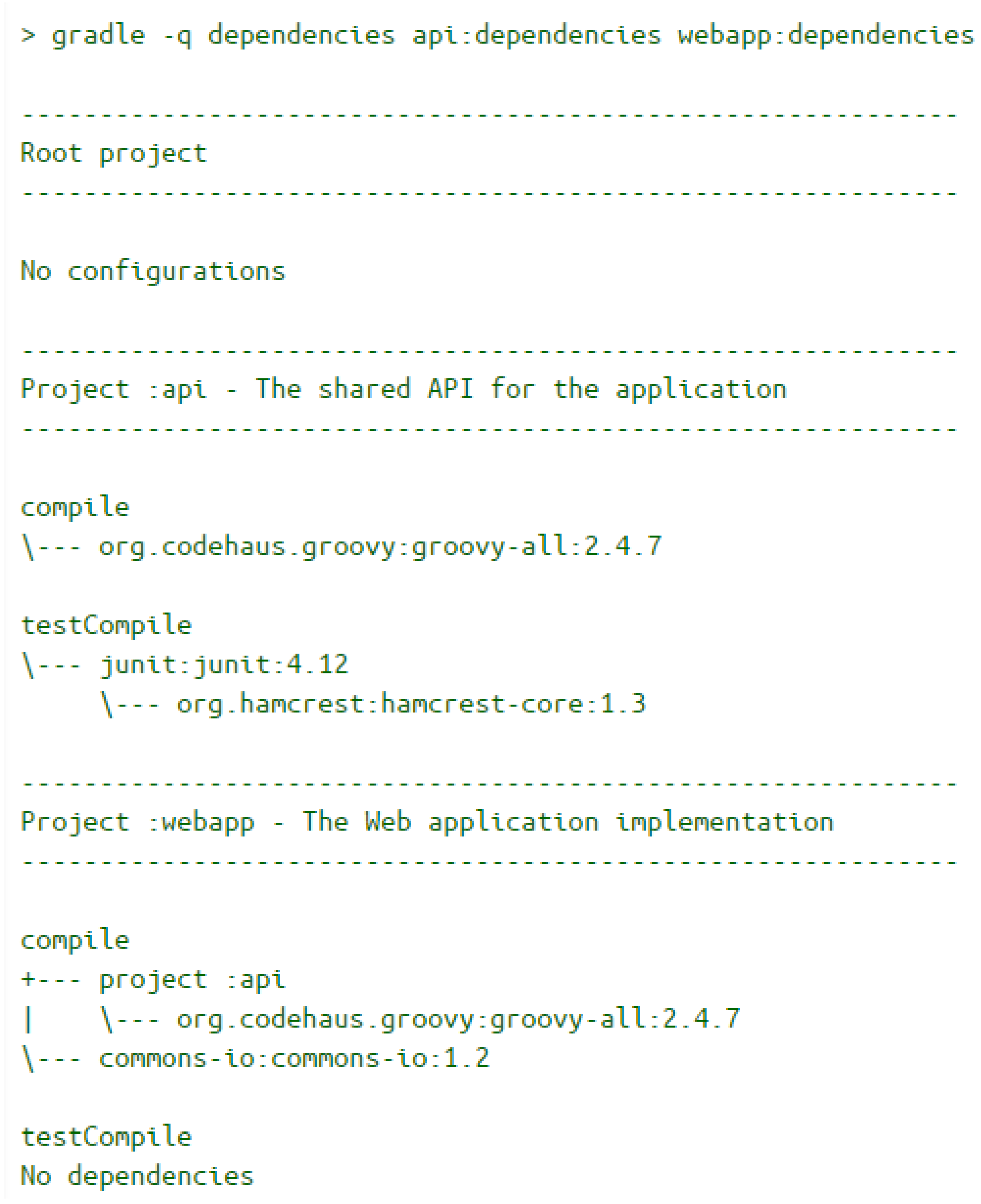
By running gradle help --task someTask gives detailed information about a specific task or multiple tasks matching the given task name in the multi-project build. Below is an example of this detailed information.



```
> gradle -q help --task libs
Detailed task information for libs

Paths
     :api:libs
     :webapp:libs

Type
     Task (org.gradle.api.Task)

Description
     Builds the JAR

Group
     build
```

Figure 23: Example of obtaining detailed help for tasks

Running gradle dependencies gives a list of the dependencies of the selected project, broken down by configuration. For each configuration, the direct and transitive dependencies of that configuration are shown in a tree. Below is an example of this report.



```
> gradle -q dependencies api:dependencies webapp:dependencies

------------------------------------------------------------
Root project
------------------------------------------------------------

No configurations

------------------------------------------------------------
Project :api - The shared API for the application
------------------------------------------------------------

compile
\--- org.codehaus.groovy:groovy-all:2.4.7

testCompile
\--- junit:junit:4.12
     \--- org.hamcrest:hamcrest-core:1.3

------------------------------------------------------------
Project :webapp - The Web application implementation
------------------------------------------------------------

compile
+--- project :api
|    \--- org.codehaus.groovy:groovy-all:2.4.7
\--- commons-io:commons-io:1.2

testCompile
No dependencies
```

Figure 24: Example of obtaining information about dependencies

Finally, the --profile command line option will record some useful timing information while your build is running and write a report to the build/reports/profile directory. The report will be named using the time when the build was run. This report lists summary times and details for both the configuration phase and task execution. The times for configuration and task execution are sorted with the most expensive operations first. The task execution results also indicate if any tasks were skipped (and the reason) or if tasks that were not skipped did no work.

Builds which utilize a buildSrc directory will generate a second profile report for buildSrc in the buildSrc/build directory.

Integrated Development Environment

Although Oracle recommends Netbeans, IntelliJ IDEAby JetBrains has been used as the main Integrated Development Environment. IntelliJ IDEA allows you to work with several projects simultaneously, each one in a separate window. The projects are independent, and cannot share information, except for the Clipboard operations. All the projects run in the same instance of IntelliJ IDEA and use the same memory space.

Comparing to the Netbeand, IntelliJ has a better functionality for optimizing and fixing imports. Also searching a dependency in Gradle is more easy from code. Finally IntelliJ has a better user interface which offers better user experience than Netbeans.

The Community edition has been selected which is free distributed and has various capabilities except of the enterprise development, like application server integration which is not needed in the particular implementation.

Chapter 5 - Analysis & Architectural Design

In this section the reader could find the analysis and architectural design of the thesis implementation. This analysis and design is demonstrated through UML diagrams, like use case, class and sequence diagrams. An architecture design tool was utilized in order the aforementioned diagrams to be created by applying the best Software Engineering practices. For every type of diagram a sub section will be dedicated which includes further and detailed description.

The analysis has been made according to the techniques that are defined by the Software Engineering theory using Visual Paradigm. The aforementioned software is an architectural design tool which provides many capabilities. A part of them is the diagram design by which the software architect can create use case, sequence, class and many other diagrams types.

The need of architectural analysis derives from the primary need for a high level definition of the system before its implementation startup as well as for the visual description of the requirements (functional mainly).

The theoretical background of the analysis is the off Desktop Interaction. This term describes the Human Computer Interaction concept from the perspective of the operating system control, which is achieved not through the traditional input devices, like mouse or keyboard. Instead modern sensor technology is being used in order to replace the aforementioned input devices which in this case is the Leap.

The above mentioned interaction could be more comprehensive through the analysis that follows in the next captions. Consequently, this is a key chapter for the reader in order to understand the off desktop concept through the architecture diagrams.

Use Case Diagrams

In this sub-section will be demonstrated the use case diagram for the thesis implementation. The below diagram aggregates four individual and independent use cases which are being placed under the umbrella of the mouse control use case scenario.

The first use case is called Cursor Movement. This particular case describes the movement of the screen cursor by the user.

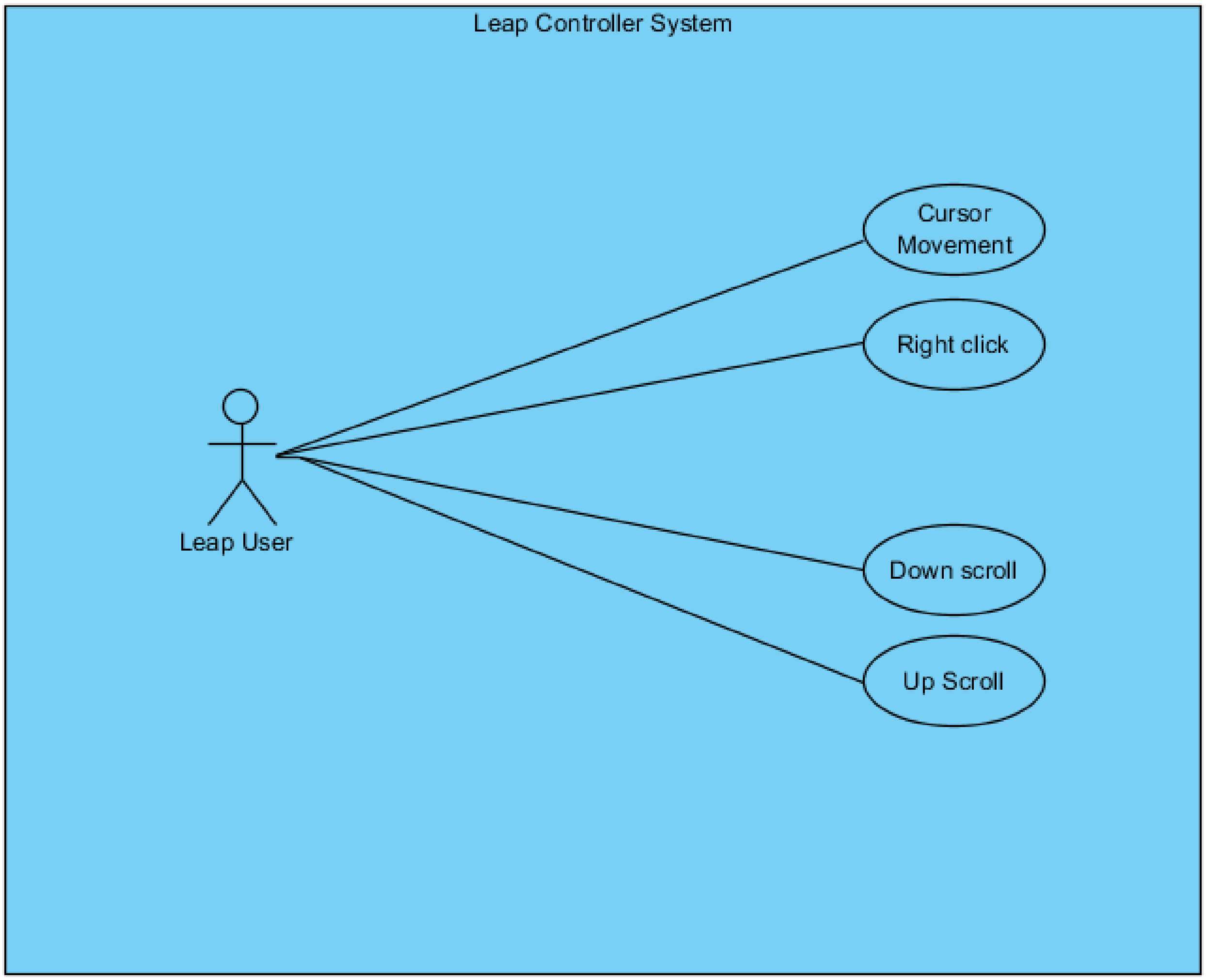


Figure 25: Use cases for Leap application

In the above diagram the reader can see five use cases which apply to the thesis' implementation. The first use case is the Cursor Movement. The action that triggers this use case is the movement of the user's hand in various directions. The application tracks the hand movement and binds the mouse cursor on it. This is the main and the most important feature of the application. The x-y-z captured coordinates are being calibrated according to the screen dimensions in order the movement of the cursor to be as accurate and precise as possible. The hand is being tracked continuously until it is out of the sensor's capture range. When the hand is being placed out of the sensor's bounds then the cursor stops its movement in the screen.

The second use case that is figured is the right click. The trigger of this case is the Tap Screen gesture the below figure shows. A screen tap gesture is recognized when the tip of a finger pokes forward and then springs back to approximately the original position, as if tapping a vertical screen. The tapping finger must pause briefly before beginning the tap according to the Leap motion.

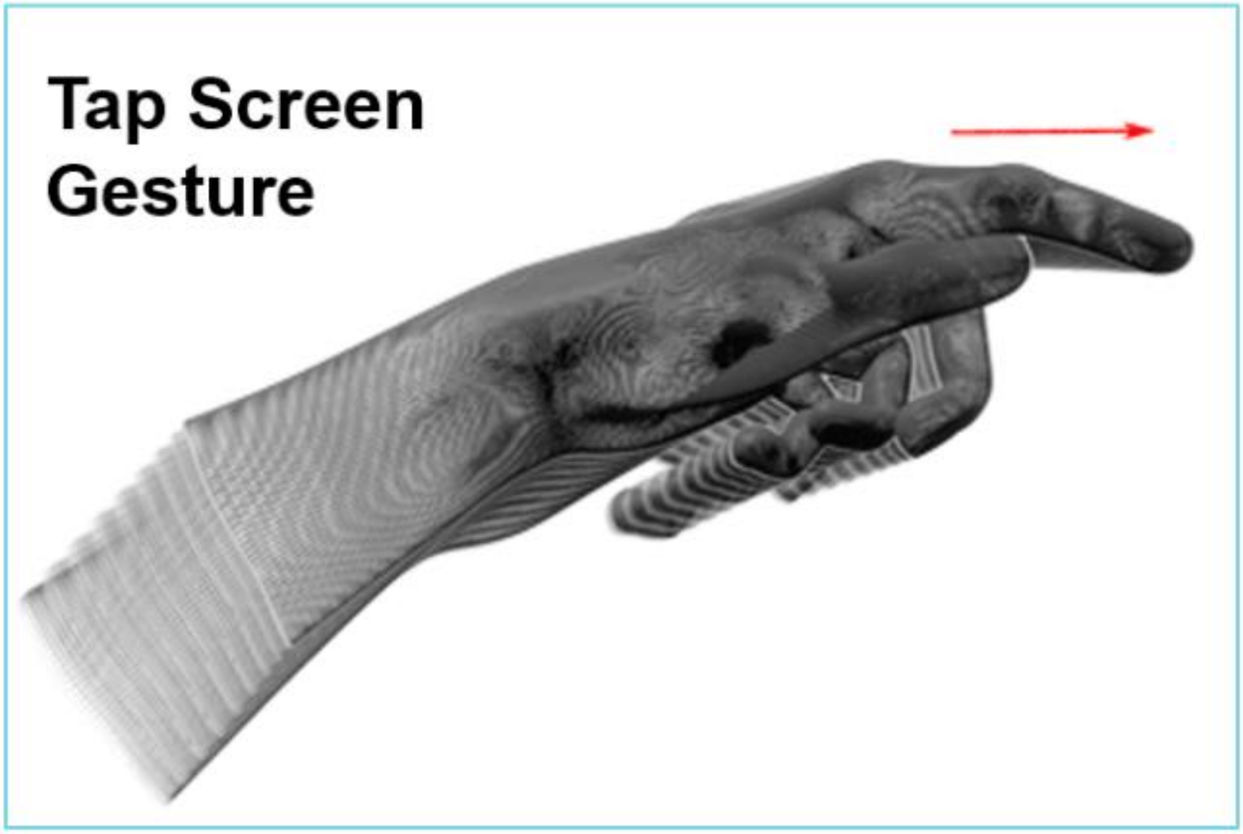


Figure 26: Tap screen gesture for Leap sensor [46]

Consequently the reader could see the down scroll use case. In this case the trigger is the gesture Clockwise Circle. Through this gesture the user can simulate the scroll up and down feature which is traditionally achieved by the wheel of the mouse. Specifically, the clockwise simulates the scroll up and the counter clockwise the scroll up. The speed of the scrolling is configurable.

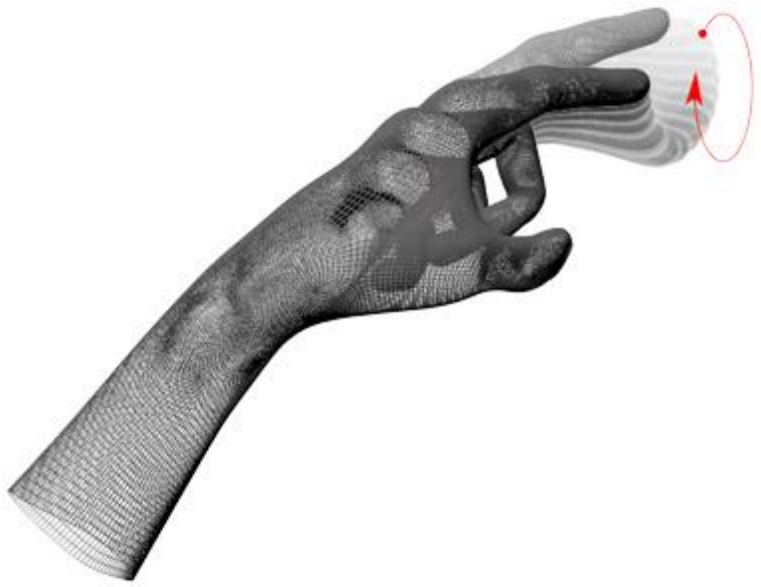


Figure 27: Clockwise circle gesture for Leap sensor [47]

Class Diagram

In this section the reader can find the class diagram which are referred to the thesis' implementation.

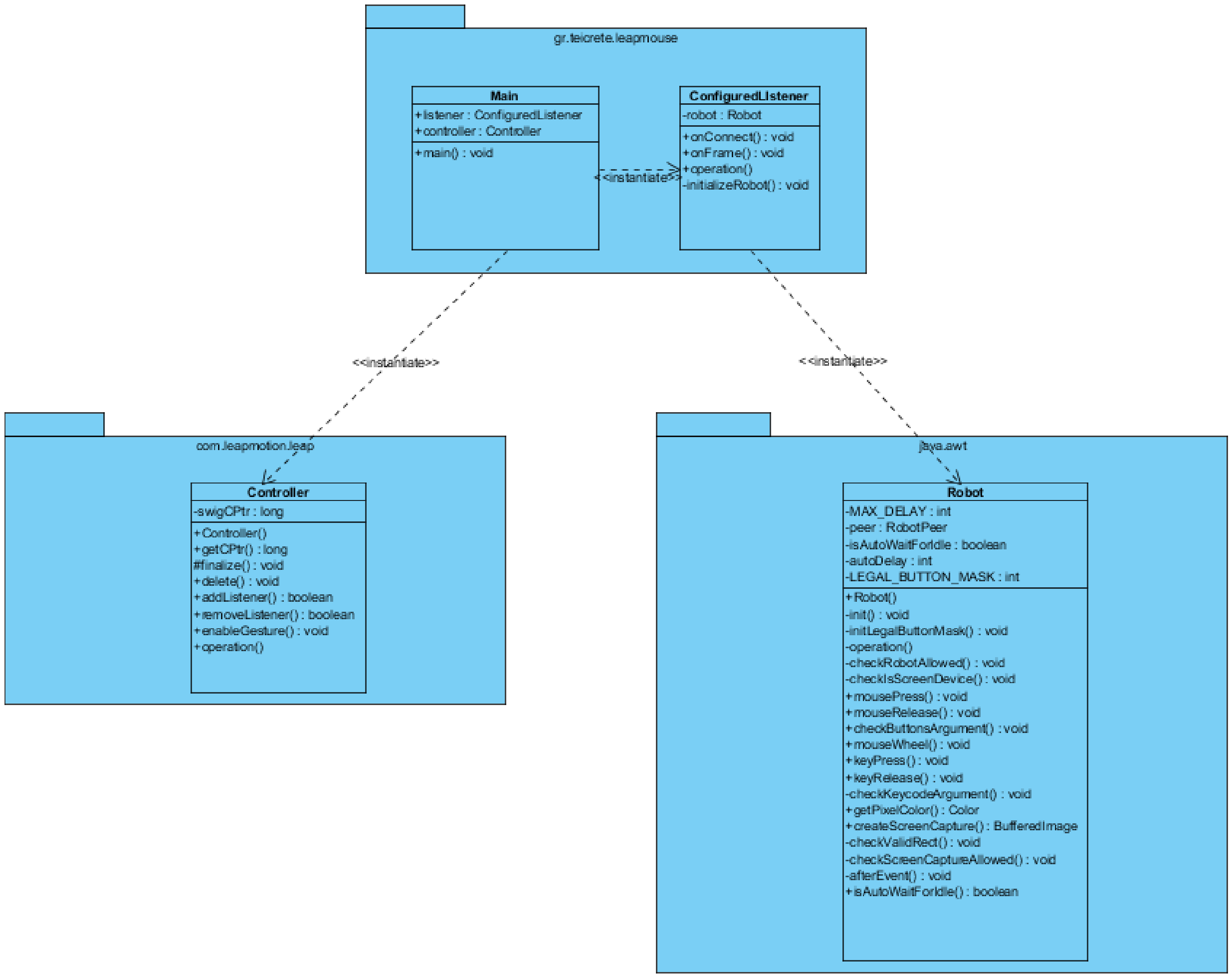


Figure 28: Class Diagram

The above class diagram demonstrates the four main classes that are used in order the application to run. The Main class instantiates the ConfiguredListener and the Controller. The Main and the Controller are part of the gr.teicrete.leapmouse package and the Controller is part of the com.leapmotion.leap package. Finally, the ConfiguredListener class instantiates the Robot class which is part of the java.awt package.

Chapter 6 - Implementation & Scenarios of Use

This chapter refers to the implementation part of the thesis project. In this section the reader can find the most important parts of the implementation code, followed by a brief description of its functionality, as well as five basic scenario that are based on the application that was developed. The code is written with Java programming language combined with all the best practices of programming and the language conventions which are defined by Oracle. Additionally screenshots of the running scenario will be demonstrated in order the reader to visualize a basic usage of application.

Code Analysis

In this sections is placed the analyses of the application's code. For brevity reasons the catholic code is not demonstrated. On the contrary only the critical and important parts are analyzed here. Should the reader investigate the catholic code, it will be available at the library of the Technological Institute of Crete.

Main Class

```
public class Main {

    public static void main (String[] args) {
        ConfiguredListener listener = new ConfiguredListener();
        Controller controller = new Controller();
        controller.addListener(listener);

        try {
            System.in.read();
        } catch (Exception e) {
            System.out.printf("Exception occured: " +
e.getMessage());
        }

    }
}
```

Figure 29: The main class of the project

The above snippet of code is the Main class of the project. It contains the main method which is the entry point of the program. As the reader could observe this class has public access modifier, static keyword and void as return type. The access modifier is public in order to permit its invocation from anywhere outside of the application's scope. Furthermore, it is defined as static because there is no Object at the time of invocation by the JVM. Its return type is void due to the no need of returning any Object.

Following the code flow the reader can understand that there are two instantiations inside main method. First an instance of ConfiguredListener class is declared, followed by an instance of the Controller class. The aforementioned class is an extension of the Java Leap API. Its operation is the configuration of the Leap sensor in order to run under certain conditions. This class will be further explained in the next pages.

Afterwards the controller Object is linked with the listener through the addListener method. Here it could be mentioned that the Controller is an integrated class of the Java Leap API.

Subsequently, inside a try-catch block the method System.in.read() is called. The "in" Object is a standard Java input stream which is open and ready to supply input data from an input source which is defined by the host environment or user. In this case the input source is the Leap sensor. System is a static class which belongs to the java.lang package. Because of its static property it cannot be instantiated. It provides three main facilities according to the Java 8 documentation, standard input, standard output and error output streams.

ConfiguredListener Class

The ConfiguredListener class contains the two basic methods of the application, the onConnect and onFrame methods. The first is shown in the below snippet.

```
public void onConnect(Controller controller) {

    controller.setPolicy(Controller.PolicyFlag.POLICY_BACKGROUND_FRAMES);
    controller.setPolicy(Controller.PolicyFlag.POLICY_IMAGES);
    controller.setPolicy(Controller.PolicyFlag.POLICY_OPTIMIZE_HMD);

    controller.enableGesture(Gesture.Type.TYPE_CIRCLE);
    controller.enableGesture(Gesture.Type.TYPE_SCREEN_TAP);
    controller.enableGesture(Gesture.Type.TYPE_SWIPE);

}
```

Figure 30: A method of ConfiguredListener Class

The above method starts by setting three policies flags. This is necessary due to the lack of default permitted policies. Firstly the reader can see the POLICY_BACKGROUND_FRAMES constant which configures the Controller in order the application to receive frames when it is not on the foreground [48]. Subsequently comes the POLICY_IMAGES flag which requests that your application receives images from the device cameras. The “Allow Images” checkbox must be enabled in the Leap Motion Control Panel or this policy will be denied [48]. Third, the reader can find the POLICY_OPTIMIZE_HMD. This particular flag request that the tracking is optimized for head-mounted tracking. The optimize HMD policy improves tracking in situations where the Leap Motion hardware is attached to a head-mounted display. This policy is not granted for devices that cannot be mounted to an HMD, such as Leap Motion controllers embedded in a laptop or keyboard [48].

Afterwards the gestures are enabled. In this case there are three types of gesture that the application needs to use in order to replace the mouse functionality. These gestures are being demonstrated in Chapter 5.

```
public void onFrame(Controller controller) {
    initializeRobot();
    Frame frame = controller.frame();
    InteractionBox box = frame.interactionBox();

    for (Finger f : frame.fingers()) {
        if (f.type() == Finger.Type.TYPE_INDEX) {
            Vector fingerPosition = f.stabilizedTipPosition();
            Vector fingerPositionBox =
            box.normalizePoint(fingerPosition);
            Dimension screenSize =
            Toolkit.getDefaultToolkit().getScreenSize();
            robot.mouseMove((int)(screenSize.width *
            fingerPositionBox.getX()),
           (int)(screenSize.height * fingerPositionBox.getY()));
        }
    }

    for (Gesture g : frame.gestures()) {
        if (g.type() == Gesture.Type.TYPE_CIRCLE) {
            CircleGesture c = new CircleGesture();
            //check if the circle is clockwise
            if (c.pointable().direction().angleTo(c.normal()) <=
            Math.PI/4) {
                robot.mouseWheel(1);
                try {
                    Thread.sleep(50);
                } catch (Exception e) {
                    System.out.println(e.getMessage());
                }
            } else /*check if the circle is counterclockwise*/ {
                robot.mouseWheel(-100);
                try {
                    Thread.sleep(50);
                } catch (Exception e) {
                    System.out.println(e.getMessage());
                }
            }
        } else if (g.type() == Gesture.Type.TYPE_SCREEN_TAP) {
            ScreenTapGesture screentapGesture = new
            ScreenTapGesture();
            robot.mousePress(InputEvent.BUTTON1_MASK);
            robot.mouseRelease(InputEvent.BUTTON1_MASK);
        }
    }
}
```

```
public void onFrame(Controller controller) {
    initializeRobot();
    Frame frame = controller.frame();
    InteractionBox box = frame.interactionBox();

    for (Finger f : frame.fingers()) {
        if (f.type() == Finger.Type.TYPE_INDEX) {
            Vector fingerPosition = f.stabilizedTipPosition();
            Vector fingerPositionBox =
            box.normalizePoint(fingerPosition);
            Dimension screenSize =
            Toolkit.getDefaultToolkit().getScreenSize();
            robot.mouseMove((int)(screenSize.width *
            fingerPositionBox.getX()),
           (int)(screenSize.height * fingerPositionBox.getY()));
        }
    }

    for (Gesture g : frame.gestures()) {
        if (g.type() == Gesture.Type.TYPE_CIRCLE) {
            CircleGesture c = new CircleGesture();
            //check if the circle is clockwise
            if (c.pointable().direction().angleTo(c.normal()) <=
            Math.PI/4) {
                robot.mouseWheel(1);
                try {
                    Thread.sleep(50);
                } catch (Exception e) {
                    System.out.println(e.getMessage());
                }
            } else /*check if the circle is counterclockwise*/ {
                robot.mouseWheel(-100);
                try {
                    Thread.sleep(50);
                } catch (Exception e) {
                    System.out.println(e.getMessage());
                }
            }
        } else if (g.type() == Gesture.Type.TYPE_SCREEN_TAP) {
            ScreenTapGesture screentapGesture = new
            ScreenTapGesture();
            robot.mousePress(InputEvent.BUTTON1_MASK);
            robot.mouseRelease(InputEvent.BUTTON1_MASK);
        }
    }
}
```

Figure 31: onFrame method of ConfiguredListener class

At the above method there are placed two main processes which can be identified by the "for loops". The first is the binding of the index finger' movement with mouse cursor. This is caused by using the method mouseMove which is a method of the Robot class retrieved from the Java Leap API. The position of the finger is being calibrated considering the size of the screen combined with the cursor's position.

The second process is the binding of the finger gestures with the three main functionalities of the mouse (right click, left click, scroll up and scroll down). The thread is being placed into sleep mode for a specific number of milliseconds in order the movement of the cursor to be slower which combined with the fact of the calibration provides better accuracy.

Leap Java API Extension

According to the official site of Leap motion, the software uses an internal model of a human hand to provide predictive tracking even when parts of a hand are not visible. The hand model always provides positions for five fingers, although tracking is optimal when the silhouette of a hand and all its fingers are clearly visible. The software uses the visible parts of the hand, its internal model, and past observations to calculate the most likely positions of the parts that are not currently visible. The subtle movements of fingers tucked against the hand or shielded from the Leap Motion sensors are typically not detectable. A Hand.confidence() rating indicates how well the observed data fits the internal model.

Every API extension is used in order to bring the capabilities of a system to a new extended bound. API boundaries could be extended in various directions depending on what functionality is needed to be added in every case. The current thesis implementation is actually an extension of the Leap motion API. The ConfiguredLisstener class is the main body of the aforementioned extension due to the augmented capabilities that it offers to every developer. Particularly, it offers the capability of the configuration regarding the policies that are necessary for the better functionality of the device. Additionally, it offers the augmented capability of cursor control through the movement of the hand and three particular gestures. This extended capability can be utilized in multimodal systems that emphasize to the natural human-computer interaction.

A practical usage of this API extension is the control of wall screens in smart houses. Also it can be used in medical science (e.g. in surgeries or general medical devices that integrate a screen). Furthermore its usage could be extended to the educational training for children as it is offers a very nice and pleasant user experience.

Scenarios of Use

In this section are demonstrated three basic scenarios based on the developed application program using the Leap as the main input device. These scenarios are the proof of concept of the current thesis implementation and they form examples of usage. Several additional scenarios could be applied in order to prove that natural user interaction has been evolved gradually into a major branch of human computer interaction. After the serious examination of these scenarios it can be concluded that the future modern operating systems will be based on natural user interaction. At this point it has to be mentioned that humanity is not yet familiar with that kind of interaction on personal computers, although the trend points to that way.

The five basic scenarios that are demonstrated in this section are the:

i. Media Player Manipulation
ii. Browser Manipulation
iii. Microsoft Word Manipulation
iv. Microsoft Power Point Manipulation
v. Windows Photo Viewer Manipulation

In the following sub sections the reader could find the analysis and presentation of the aforementioned scenarios through figures that display the gestures as well as the result of the interaction. The gestures that follow are universal and applied to all of the above five scenarios, so they will be demonstrated only once and mentioned through the figures for the shake of brevity.

Media Player Manipulation

In this subsection the Microsoft Media Player manipulation is being demonstrated. The two sub scenarios are:

i. Play button press
ii. Menu open

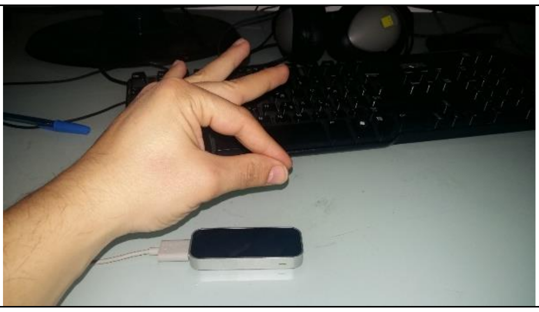


Figure 32 A: Left click gesture

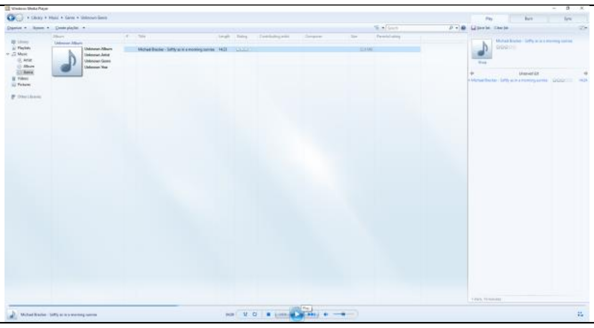
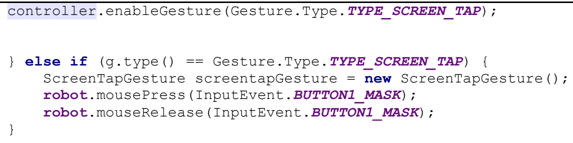


Figure 32 B: The play button is being pressed after the performed gesture



```
controller.enableGesture(Gesture.Type.TYPE_SCREEN_TAP);


} else if (g.type() == Gesture.Type.TYPE_SCREEN_TAP) {
    ScreenTapGesture screentapGesture = new ScreenTapGesture();
    robot.mousePress(InputEvent.BUTTON1_MASK);
    robot.mouseRelease(InputEvent.BUTTON1_MASK);
}
```

Figure 32 C:Left click gesture code

The above images demonstrate the scenario of a simple button pressing without keyboard or mouse usage. Initially, the user points to the button while moving his hand over the leap sensor to the direction that he wants to place the cursor (over the button). This is a continuous movement of the hand, so it cannot be understood as a gesture. Afterwards the user needs to give the command in order the button to be pressed and the music to start. The play command is given by the gesture as demonstrated in 32 A figure.

The aforementioned gesture is a movement combination of the thumb and index finger. At this point it has to be mentioned that the particular gesture has an approximate 90 percent of successfulness due to the sensor sensibility and accuracy. The algorithm has been tested and no exception throwing have been detected.

In figure 32 B is demonstrated the result of the gesture which is the fact that the player starts the song which is selected after the gesture has been performed. In fact the media player is controlled through the gesture that the user performs and the result is one the application’s main functionalities. It has to be mentioned that this application has many features and capabilities that could be supported by a wider set of gestures.

In figure 32 C is demonstrated the code of the left click gesture. Firstly the gesture is enabled by the first line of code. Subsequently comes the snippet of code that checks if the gesture is of a particular type in order to perform the appropriate and desired action, the left click. This is done by calling two methods of the Robot class, the mouse press and mouse release. Actually this is a programmatical simulation which is triggered by the aforementioned gesture, since the action which is performed with the established on desktop way (using the mouse input device) is similar. The user presses down the left button of the mouse and releases it afterwards.

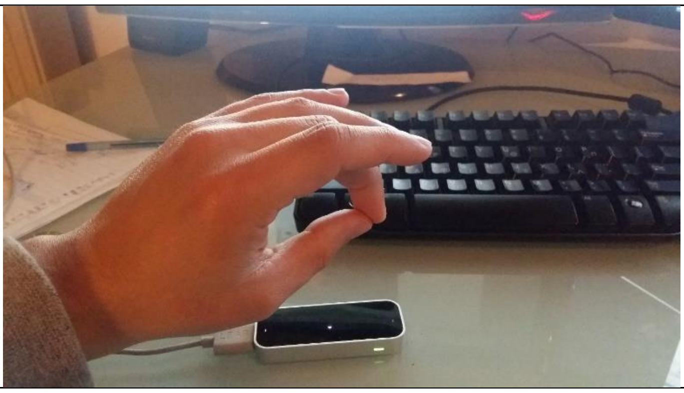


Figure 33 A: Right click gesture

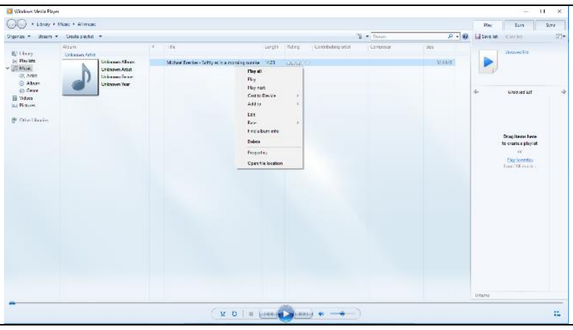


Figure 33 B: The menu opens after the performed gesture

```
controller.enableGesture(Gesture.Type.TYPE_SCREEN_TAP);


} else if (g.type() == Gesture.Type.TYPE_SCREEN_TAP) {
    ScreenTapGesture screentapGesture = new ScreenTapGesture();
    robot.mousePress(InputEvent.BUTTON2_MASK);
    robot.mouseRelease(InputEvent.BUTTON2_MASK);
}
```

Figure 33 C: Right click gesture code

The above images demonstrate the scenario of a Windows Media Player menu opening without keyboard or mouse usage. Initially, the user points to the button while moving his hand over the leap sensor to the direction that he wants to place the cursor. This is a continuous movement of the hand, so it cannot be understood as a gesture. Afterwards the user needs to give the command in order the menu to open. This command is given by the gesture that is demonstrated the 33 A figure.

The aforementioned gesture is a movement combination of the thumb and middle finger. At this point it has to be mentioned that the particular gesture has an approximate 90% of successfulness due to the sensor sensibility and accuracy.

Figure 33 B demonstrates the result of the gesture that is the fact that the menu opens in order the user to select one of the available options. Afterwards the user can select the desired action by using the left click gesture as demonstrated in the above sub section.

Figure 33 C shows the implementation code which is a programmatical simulation of the right click gesture. In the first line of code the desired gesture is enabled in order to be trackable. Subsequently happens the simulation by calling the mousePress and mouseRelease methods of the robot class. These methods are called with the BUTTON2_MASK event that correspond to the right button of the mouse.

Browser Manipulation

In this section will be demonstrated a minimal browser manipulation scenario. This scenario aggregates three sub scenarios which are:

i. Click and open a bookmark
ii. Open the browser menu
iii. Scroll up

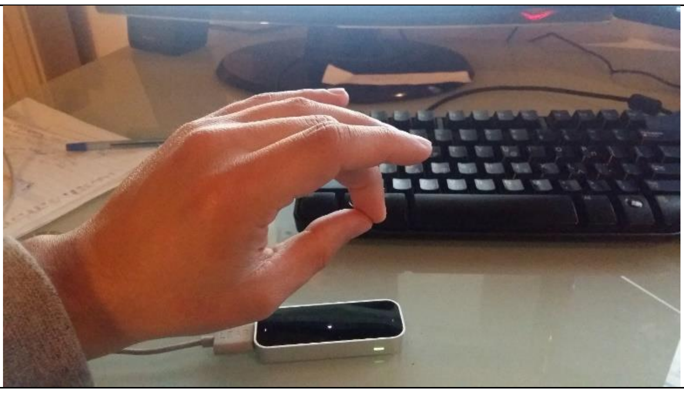


Figure 34 A: Right click gesture

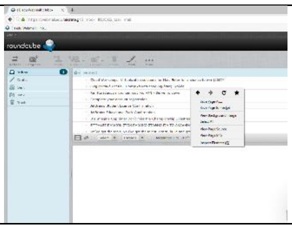


Figure 34 B: The user performs the gesture in order to open the browser's bookmark

```
controller.enableGesture(Gesture.Type.TYPE_SCREEN_TAP);


} else if (g.type() == Gesture.Type.TYPE_SCREEN_TAP) {
    ScreenTapGesture screentapGesture = new ScreenTapGesture();
    robot.mousePress(InputEvent.BUTTON1_MASK);
    robot.mouseRelease(InputEvent.BUTTON1_MASK);
}
```

Figure 34 C: Right click gesture code

The 34 B figure shows the result of the scenario which happens after clicking to the browser's bookmark through the gesture that is shown in the Figure 32. The bookmark links to the web mail client of the Technological Institute of Crete. The next step is to open the browser's menu through the right click gesture.

The above figure shows the action that is performed by the user through the right click gesture. This gesture is displayed in figure 34 A and is described exhaustively in the above sub section. The result of this action is that the main context menu opens and the user can select one of its options through the left click gesture.

The code snippet is the programmatical simulation of the right click gesture that is described in the previous section.

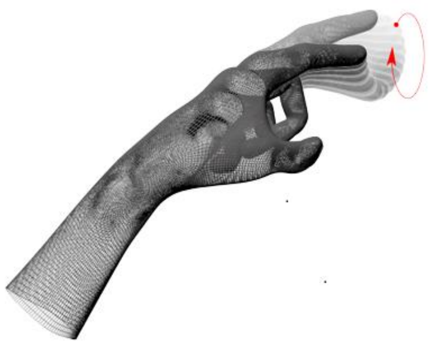
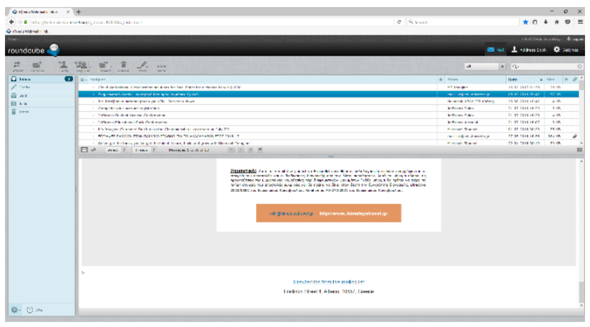


Figure 35 A: Circle gesture with the forefinger [52]



```
controller.enableGesture(Gesture.Type.TYPE_CIRCLE);

if (g.type() == Gesture.Type.TYPE_CIRCLE) {
    CircleGesture c = new CircleGesture();
    //check if the circle is clockwise
    if (c.pointable().direction().angleTo(c.normal()) <= Math.PI/4) {
        robot.mouseWheel(1);
        try {
            Thread.sleep(50);
        } catch (Exception e) {
            System.out.println(e.getMessage());
        }
    } else /*check if the circle is counterclockwise*/ {
        robot.mouseWheel(-100);
        try {
            Thread.sleep(50);
        } catch (Exception e) {
            System.out.println(e.getMessage());
        }
    }
```

Figure 35 B: Scroll down result

Figure 35 C: Circle gesture code

Figure 35 B shows the result of the scroll down gesture. The user scrolls down to the content of the selected email.

Figure 35 A shows the circle gesture which is performed with the forefinger, the scroll down gesture. This is a clockwise movement. The opposite movement (counter clockwise) will have as a result the scrolling up action. For the shake of brevity this action is not demonstrated.

Figure 35 C demonstrates the code that enables the Circle gesture as well as tracks this particular movement. The code also checks if the gesture is clockwise or counterclockwise in order to perform the right action which in this case is the scroll down. In case of an Exception the error is logged to the console. If this snipper of code is integrated into a larger application then the logging could be performed on a log file instead of the console.

Microsoft Word Manipulation

In this section will be demonstrated a minimal browser manipulation scenario. This scenario aggregates three sub scenarios which are:

i. Select font
ii. Open application’s menu
iii. Scroll down to the page

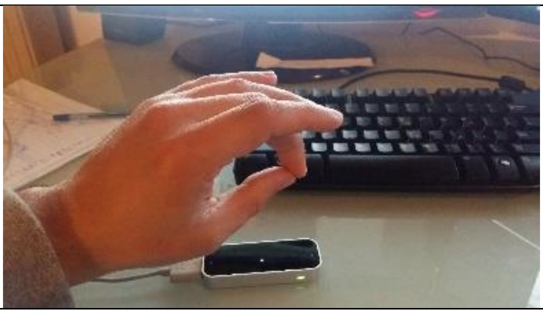
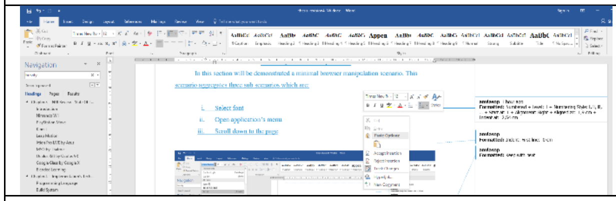


Figure 36 A: Right click gesture



In this section will be demonstrated a minimal browser manipulation scenario. This scenario aggregates three sub scenarios which are:

i. Select font
ii. Open application's menu
iii. Scroll down to the page

Figure 36 B: Open content menu action

```
controller.enableGesture(Gesture.Type.TYPE_SCREEN_TAP);

} else if (g.type() == Gesture.Type.TYPE_SCREEN_TAP) {
    ScreenTapGesture screentapGesture = new ScreenTapGesture();
    robot.mousePress(InputEvent.BUTTON1_MASK);
    robot.mouseRelease(InputEvent.BUTTON1_MASK);
}
```

Figure 36 C: Right click gesture code

The first step of the current scenario is the font selection. The result of this step is demonstrated in the figure 36 B. In this particular step the user points to the Font menu item through his hand movement over the Leap. After the user has reached the desired menu item he proceeds to the left click gesture as it shown in figure 36 B. Finally, after he finds the desired font to select he proceeds again to the same gesture (left click) in order to select it.

The second step of the current scenario is to open the content menu of Word application. For this action the user makes the right click gesture which is shown in figure 36 A. Afterwards he can point to the preferred option using his hand movement over the Leap sensor and select one of them by making the left click gesture as demonstrated in the previous step.

Microsoft Power Point Manipulation

In this sub section will be demonstrated the creation of a new Microsoft Power Point document slide for presentation purposes. This goal will be accomplished through the aforementioned gestures (hand bind to cursor, right click and left click gestures). The particular process includes the following actions: slide creation and slide demonstration during a speech.

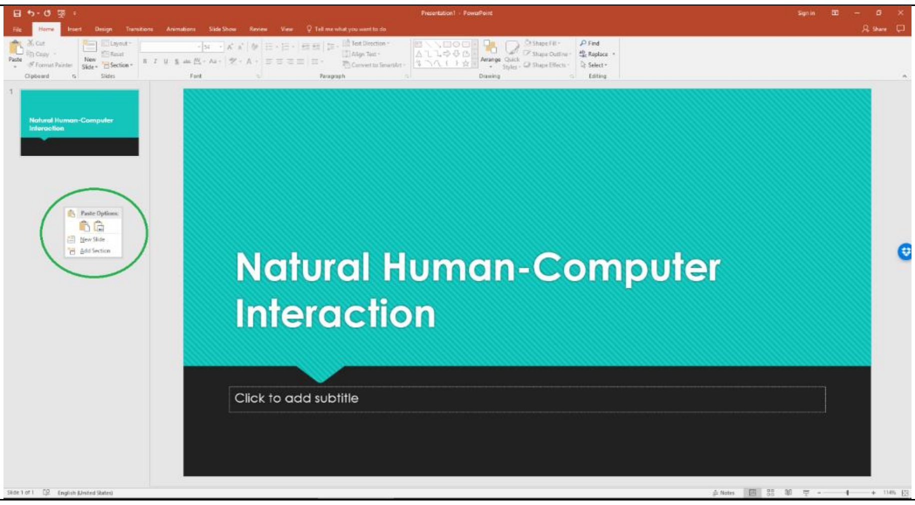

Natural Human-Computer Interaction
Paste Options
New Slide
Add Section
Natural Human-Computer Interaction
Click to add subtitle

Figure 37 A: Open menu - right click gesture

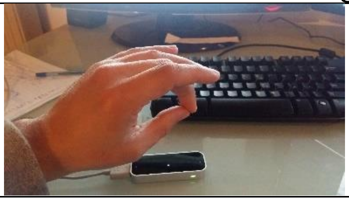


Figure 37 B: right click gesture

```
controller.enableGesture(Gesture.Type.TYPE_SCREEN_TAP);


} else if (g.type() == Gesture.Type.TYPE_SCREEN_TAP) {
    ScreenTapGesture screentapGesture = new ScreenTapGesture();
    robot.mousePress(InputEvent.BUTTON1_MASK);
    robot.mouseRelease(InputEvent.BUTTON1_MASK);
}
```

Figure 37 C: right click gesture code

The above three figures demonstrate the process of a new slide creation within Microsoft PowerPoint document. The first goal step is accomplished through the right click gesture. After the aforementioned gesture the menu opens and the available menu options are displayed.

Afterwards the user uses the bind to cursor hand movement gesture in order to point the cursor to the desired menu option which is the “new slide”. This option is highlighted by PowerPoint application and corresponds to the creation of a new document slide. Now the user is ready to proceed to the next step which is the left click gesture.

Windows Photo Viewer Manipulation

In this sub section is demonstrated a photo viewer manipulation. The story of this section is the navigation (front and back) through images as well as zoom in and zoom out of a particular photo using the leap sensor and the aforementioned gestures. An example of use of this feature is a magnetic resonance manipulation by a doctor through touchless gestures. The benefit of this process is that the doctor will maintain his sterilization.

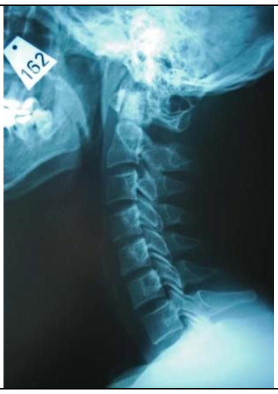


Figure 38 A: Magnetic resonance in its original size

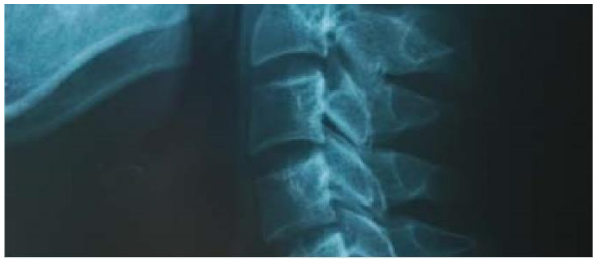


Figure 38 B: Magnetic resonance zoomed in

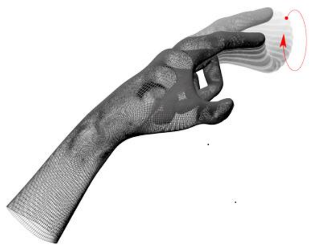
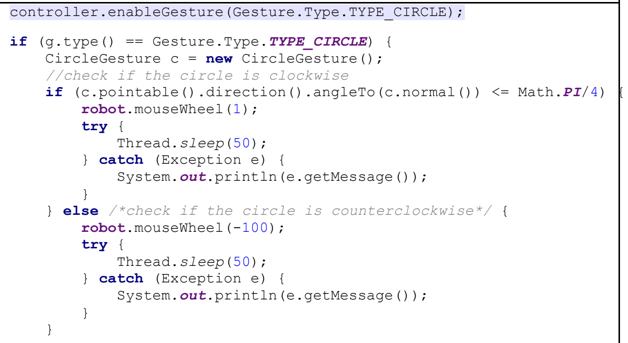


Figure 38 C: Circle gesture with the forefinger [52]



```
controller.enableGesture(Gesture.Type.TYPE_CIRCLE);

if (g.type() == Gesture.Type.TYPE_CIRCLE) {
    CircleGesture c = new CircleGesture();
    //check if the circle is clockwise
    if (c.pointable().direction().angleTo(c.normal()) <= Math.PI/4) {
        robot.mouseWheel(1);
        try {
            Thread.sleep(50);
        } catch (Exception e) {
            System.out.println(e.getMessage());
        }
    } else /*check if the circle is counterclockwise*/ {
        robot.mouseWheel(-100);
        try {
            Thread.sleep(50);
        } catch (Exception e) {
            System.out.println(e.getMessage());
        }
    }
```

Figure 38 D: Circle gesture code

Figure 38 A shows a magnetic resonance in its original size. The desired doctor's action is to zoom this image in order to examine more carefully a specific area of the tomography. This is accomplished through the scroll-in gesture which is displayed in Figure 38 C. The result of this action is demonstrated in Figure 38 B where the reader can see that the magnetic resonance is zoomed in to the desired area of focus. With the reverse gesture (counter clockwise) the doctor can zoom out the image or restore it to the original size.

At this point it has to be mentioned that the application that is used in order to display the image is Windows Photo Viewer, nevertheless this process could be applied in several other specialized applications which are dedicated for medical purposes after meeting the condition of supporting the aforementioned features (zoom and image navigation).

Table 1: Comparative table of interaction – Leap/Mouse

Benefits of Interaction	The Leap	Mouse
Sterilization Maintenance	✓	✗
High Accuracy	✗	✓
Satisfaction of Natural Interaction	✓	✗
Self Control (no assistant required)	✓	✗
Limited usage training needed	✓	✗

The above table show the benefits of each interaction after a comparative analysis for each type. It can be concluded that the interaction which is based on the Leap sensor as an input device includes more benefits than the traditional way of mouse interaction.

More specifically, the greatest advantage is that through natural user interaction the doctor maintain his sterilization which is very important in several medical processes. Additionally, it is observed a higher level of user satisfaction through the aforementioned type of interaction, as well as a limited need for usage training. The mouse prevails to the matter of high accuracy, nevertheless this can be overridden in the future by higher accuracy sensors combined with the appropriate algorithms.

As a conclusion it can be drawn that interaction with Leap offers several benefits to the doctor which does not exist using the classic way of mouse interaction.

Chapter 7 - Conclusions & Future Work

The comprehension of interactive relation between human and computer as well as the means of its attainment, which constitutes the major goal of this thesis, arising from the study of the 4, 5, 6 chapters in which UML diagrams, programming code, scenarios as well as technologies that were used are cited.

More specifically, in chapter 4 are cited in a detailed way the technical means and the environment which were selected as appropriate and widely prevalent in use for the code development process.

In chapter 5 is presented the preparative stage for the implementation of thesis' application, through the apposition of two individual diagrams. In the first diagram (figure 25) are analytically cited the individual capabilities of the current application for individual, particular and daily operations of a computer user, so as the reader to perceive its usefulness. In the second (figure 28) diagram a reference of class relationship is made, which led to the development of the code, in order the software engineer to check as well as to understand the exact way of implementation.

Finally in chapter 6 are demonstrated the code analysis as well as five basic scenarios. These scenarios are proofs of concepts which combine the theoretical part with the implementation,

It is obvious that all the above form the core and the minimum of work required to implement the current application. Apparently, the exponential evolution of technology both in hardware and the software as well as each software engineer's personal objectives and needs enabling the improvement of the specific application but also its use extension in other software applications.

References

[1] Devine, P. G., & Sherman, S. J. (1992). Intuitive versus rational judgment and the role of stereotyping in the human condition: Kirk or Spock? Psychological Inquiry, 3(2), 153-159.

[2] Hodges, F. M. (2003). The promised planet: Alliances and struggles of the gerontocracy in American television science fiction of the 1960s. The Aging Male, 6(3), 175-182.

[3] James, N. E. (1988). Two sides of paradise: The Eden myth according to Kirk and Spock. In D. Palumbo (Ed.), Spectrum of the fantastic (pp. 219-223). Westport, CT: Greenwood.

[4] Curtis, B. (2011). Natural User Interface – The Second Revolution in Human/Computer Interaction.

[5] `1q, Martin H. "The ENIAC Story". O R D N A N C E. 708 Mills Building - Washington, DC: American Ordnance Association (January–February 1961). Retrieved 29 March 2015

[6] Kim, M. (2016, February 11). Retrieved from phillyvoice: http://www.phillyvoice.com/70-years-ago-six-philly-women-eniac-digital-computer-programmers/

[7] (2011, December 13). Retrieved from ftparmy: http://ftparmy.com/243960-mirbsd-korn-shell-r40d.html

[8] Shirriff, K. (2016, June 6). Retrieved from arstechnica: https://arstechnica.com/gadgets/2016/06/y-combinators-xerox-alto-restoring-the-legendary-1970s-gui-computer/

[9] Huckaby, K. (2013, December 9). Retrieved from eecatalog: http://eecatalog.com/digital-signage/2013/12/09/the-engaging-user-experience-the-natural-user-interface/

[10] Allen, Danny (November 17, 2006). A Closer Look at the Nintendo Wii. PC World. Archived from the original on February 5, 2008. Retrieved March 8, 2007.

[11] JC Lee, (July 15, 2008). "Hacking the Nintendo Wii remote"

[12] McFerran, D. (2013, October 4). Retrieved from nintendolife: http://www.nintendolife.com/news/2013/10/pandoras_tower_creator_ganbarion_is_co_developing_wii_fit_u

[13] B., S. (2010, March 11). *Sony reveals what makes PlayStation Move tick*. Retrieved from Gamespot: http://www.gamespot.com/news/sony-reveals-what-makes-playstation-move-tick-6253435

[14] Mikhailov, A. (2009, August 31). PlayStation Motion Controller Interview Part 2. Foster City, California.

[15] Marks, R. (2001, August). Retrieved from Lukasz: http://lukasz.dk/mirror/research-scea/research/pdfs/STEFpamphlet2001.pdf

[16] (2012, September 12). Retrieved from madfanboy: http://madfanboy.com/chto-nuzhno-sdelat-sony-chtoby-spasti-ps-move-vita-i-ps4-ot-polnogo-provala

[17] Hoiem, D. (2011, December 1). *How the Kinect Works*. Retrieved from https://courses.engr.illinois.edu/cs498dh/fa2011/lectures/Lecture%2025%20-%20How%20the%20Kinect%20Works%20-%20CP%20Fall%202011.pdf

[18] Stark, C. (2012, November 29). This Is How Microsoft's Kinect Actually Works. Retrieved from Mashable: http://mashable.com/2012/11/29/microsoft-kinect/#qZ0t4mMqZkqE

[19] Microsoft. (2016). *Kinect Fusion*. Retrieved from msdn.microsoft.com: https://msdn.microsoft.com/en-us/library/dn188670.aspx

[20] Zalevsky, Z. (2007). USA Patent No. 000335.

[21] Khoshelham. (2011). Accuracy analysis of kinect depth data. ISPRS Calgary 2011 Workshop. Calgary.

[22] Zalevsky, Z et al. (2006). *USA Patent No. 8400494.*

[23] Sprung, A. (2012, February 1). Retrieved from blogs.microsoft: https://blogs.microsoft.com/newengland/2012/02/01/v1-0-kinect-for-windows-sdk-now-available/

[24] Serrano, A. (2012, September 27). Retrieved from trendhunter: https://gr.pinterest.com/pin/308355905711567928/

[25] (2016, December 1). Retrieved from community.leapmotion: https://community.leapmotion.com/t/linear-algebra-trying-to-locate-a-point-in-space-stumped-trying-to-debug/5824/3

[26] Retrieved from ingmulti: http://www.ingmulti.com/store/index.php?id_product=34&controller=product

[27] Colgan, A. (2014, August 9). *How Does the Leap Motion Controller Work?* Retrieved from http://blog.leapmotion.com/: http://blog.leapmotion.com/hardware-to-software-how-does-the-leap-motion-controller-work/

[28] (2015, August 27). Retrieved from medium: https://medium.com/@LeapMotion/how-does-the-leap-motion-controller-work-9503124bfa04#.t1nf6pvkp

[29] Asus. (n.d.). Xtion PRO. Retrieved from http://www.asus.com/: http://www.asus.com/3D-Sensor/Xtion_PRO/

[30] Retrieved from asus: https://www.asus.com/sa-en/3D-Sensor/Xtion_PRO_LIVE/

[31] Silver, C. (2015, 12 25). Gift This, Not That: Myo Armband vs. This Toaster. Retrieved from www.forbes.com: http://www.forbes.com/forbes/welcome/?toURL=http://www.forbes.com/sites/curtissilver/2015/12/23/gift-this-not-that-myo-armband-vs-this-toaster/&refURL=https://en.wikipedia.org/&referrer=https://en.wikipedia.org/

[32] Retrieved from http://philippines.liketimes.me/item/I-PA1db0d759df041f3cd2

[33] Oculus. (2015, June 11). www.oculus.com. Retrieved from The Oculus Rift, Oculus Touch, and VR Games at E3: https://www3.oculus.com/en-us/blog/the-oculus-rift-oculus-touch-and-vr-games-at-e3/

[34] Westaway, L. (2014, October 13). www.cnet.com. Retrieved from Virtual reality and the silver screen: A match made in heaven: https://www.cnet.com/news/virtual-reality-and-the-silver-screen-a-match-made-in-heaven/

[35] (2016, December 1). Retrieved from tay.kinja: http://tay.kinja.com/not-everyone-can-vr-yet-1751925206

[36] Controversy grows over Google's Glass project. (2013, March 27). Retrieved from http://www.thehindubusinessline.com/: http://www.thehindubusinessline.com/info-tech/controversy-grows-over-googles-glass-project/article4553860.ece

[37] Federal Trade Commission. (n.d.). *What We Do*. Retrieved from www.ftc.gov.

[38] Davies, C. (2013, July 5). Glass Boutique app store, MP3 player, Lock-screen & more revealed. Retrieved from http://www.slashgear.com/: http://www.slashgear.com/glass-boutique-app-store-mp3-player-lock-screen-more-revealed-05289182/

[39] US Police issues warnings for Google Glass Users over Muggery. (2014, April 21). Retrieved from http://news.biharprabha.com/: http://news.biharprabha.com/2014/04/us-police-issues-warnings-for-google-glass-users-over-muggery/

[40] (2013, April 25). Retrieved from cnet: https://www.cnet.com/products/google-glass/preview/

[41] Retrieved from http://www.quixote.com/how-google-glass-works/

[42] Friesen, N. (2012). Blended Learning.

[43] Curtis, B. J. (2006). The handbook of blended learning: Global perspectives, local designs. San Francisco: Pfeiffer.

[44] Singh, H., & Reed, C. (2001). A White Paper: Achieving Success with Blended Learning. http://www.leerbeleving.nl/.

[45] Garrison, R. (2004). Blended learning: Uncovering its transformative potential in higher education. ScienceDirect, 95–105.

[46] Retrieved from developer.leapmotion: https://developer.leapmotion.com/documentation/v2/csharp/devguide/Leap_Gestures

[47] Retrieved from https://developer.leapmotion.com/documentation/csharp/devguide/Leap_Gestures.html

[48] Retrieved from Leap Motion / Developer: https://developer.leapmotion.com/documentation/cpp/api/Leap.Controller.html

[49] Donovan, J., & Brereton, M. (2005). Movements in Gesture Interfaces. CRITICAL COMPUTING 2005 – Between Sense and Sensibility, The Fourth De cennial Aarhus Conference (pp. 6-10). Aarhus: Interaction design and Work Practice Lab. Retrieved from http://archives.ashesi.edu.gh/V3_2004_2010/RESEARCH/RESEARCH/LARSSEN/MovementBasedInteraction_AarhusWorkshopProceedings.pdf#page=14

[50] Oracle. (2015, April 14). *Oracle SE Development Kit 7 Downloads*. Retrieved from Oracle Web site: http://www.oracle.com/technetwork/java/javase/downloads/jdk7-downloads-1880260.html

[51] Git. (2016, December 1). *Git*. Retrieved from Git: https://git-scm.com/download/win

[52] Retrived from Leap Motion / Developer https://developer.leapmotion.com/documentation/csharp/devguide/Leap_Gestures.html

[53] JetBrains. (2016, November 22). JetBrains. Retrieved from IntelliJ IDEA: https://www.jetbrains.com/idea/?fromMenu

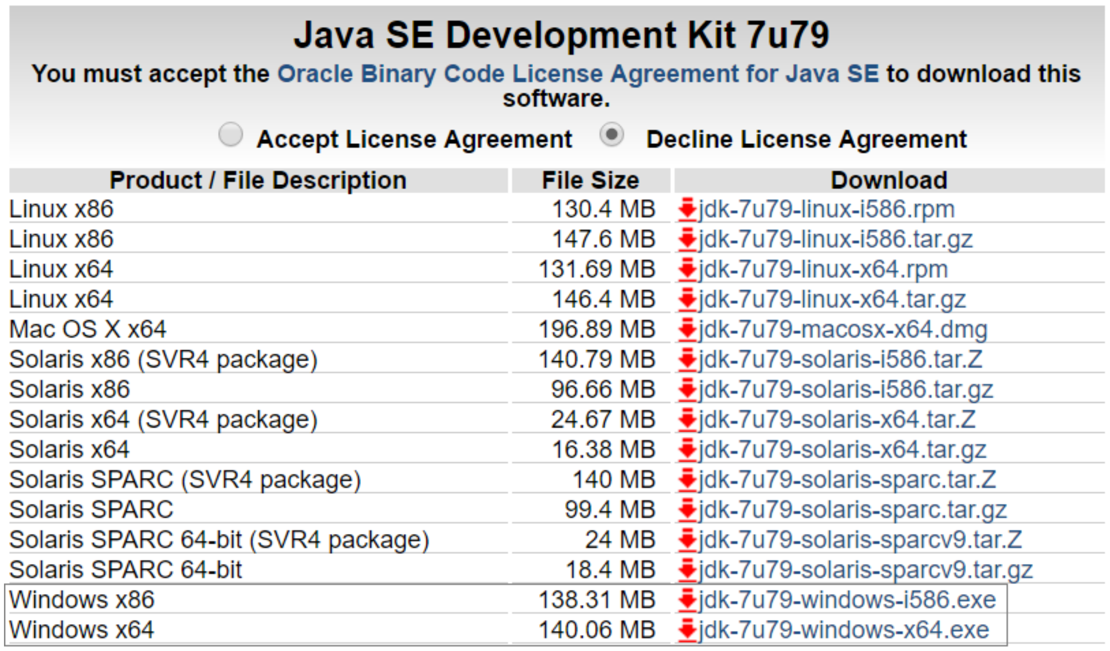
Appendix A: Environment Configuration & Integration Instructions

This chapter demonstrate the installation and setup process of the technologies and environment through detailed instruction guides. The source of the below researched information are the official sites of each technology provider.

Java

Initially, the JDK 7 should be installed locally to the environment where the application will be running. To proceed to this action the engineer should download the JDK execution file from the official site of Oracle[50]. The below guide demonstrates this action.

Step 1: After License Agreement is accepted download the Java SE Development Kit. Download the file with the "Windows x64" description.



Java SE Development Kit 7u79

You must accept the Oracle Binary Code License Agreement for Java SE to download this software.

Accept License Agreement [✓ Decline License Agreement]

Product / File Description	File Size	Download
Linux x86	130.4 MB	jdk-7u79-linux-i586.rpm
Linux x86	147.6 MB	jdk-7u79-linux-i586.tar.gz
Linux x64	131.69 MB	jdk-7u79-linux-x64.rpm
Linux x64	146.4 MB	jdk-7u79-linux-x64.tar.gz
Mac OS X x64	196.89 MB	jdk-7u79-macosx-x64.dmg
Solaris x86 (SVR4 package)	140.79 MB	jdk-7u79-solaris-i586.tar.Z
Solaris x86	96.66 MB	jdk-7u79-solaris-i586.tar.gz
Solaris x64 (SVR4 package)	24.67 MB	jdk-7u79-solaris-x64.tar.Z
Solaris x64	16.38 MB	jdk-7u79-solaris-x64.tar.gz
Solaris SPARC (SVR4 package)	140 MB	jdk-7u79-solaris-sparc.tar.Z
Solaris SPARC	99.4 MB	jdk-7u79-solaris-sparc.tar.gz
Solaris SPARC 64-bit (SVR4 package)	24 MB	jdk-7u79-solaris-sparcv9.tar.Z
Solaris SPARC 64-bit	18.4 MB	jdk-7u79-solaris-sparcv9.tar.gz
Windows x86	138.31 MB	jdk-7u79-windows-i586.exe
Windows x64	140.06 MB	jdk-7u79-windows-x64.exe

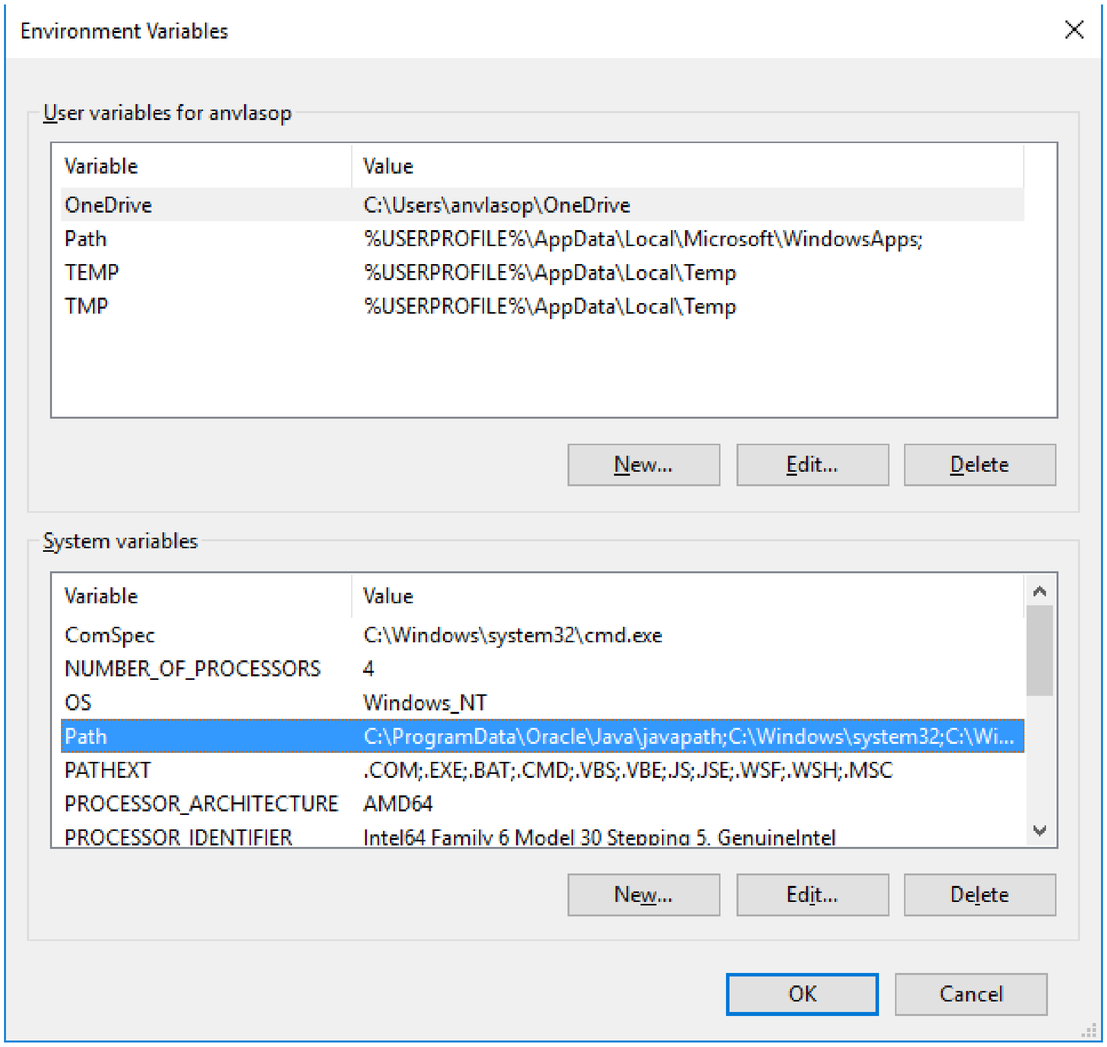
Figure 39: Available JDK versions

Step 2: Run the executable file as administrator and complete the installation of the JDK.

Step 3: Now it is time to setup the path environment variables for Java. Go to the Control Panel > System. Click the Advanced System Settings link in the left column. In the System

Properties window, click on the Advanced tab, then click the "Environment Variables" button near the bottom of that tab. In the "System Variables" section create a new variable with name "JAVA_HOME" and value the path where the jdk is installed. The path should be like the following: C:\Program Files\Java\jdk1.8.0_111. Then edit the "path" system variable and add the following value: %JAVA_HOME%\bin.

To check that everything has been setup correctly open a command line window and type "java -version". The version of the declared Java should be printed in the console. If so then the java environment path is configured successfully.


Environment Variables

User variables for anvlasop

Variable	Value
OneDrive	C:\Users\anvlasop\OneDrive
Path	%USERPROFILE%\AppData\Local\Microsoft\WindowsApps;
TEMP	%USERPROFILE%\AppData\Local\Temp
TMP	%USERPROFILE%\AppData\Local\Temp

System variables

Variable	Value
ComSpec	C:\Windows\system32\cmd.exe
NUMBER_OF_PROCESSORS	4
OS	Windows_NT
Path	C:\ProgramData\Oracle\Java\javapath;C:\Windows\system32;C:\Wi...
PATHEXT	.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH;.MSC
PROCESSOR_ARCHITECTURE	AMD64
PROCESSOR_IDENTIFIER	Intel64 Family 6 Model 30 Stepping 5, GenuineIntel

Figure 40: Environment Variables

GitHub

GitHub is a web based repository for Git. For cloning and downloading the thesis project, the engineer should create an account on the GitHub website. The following steps should be taken for this action.

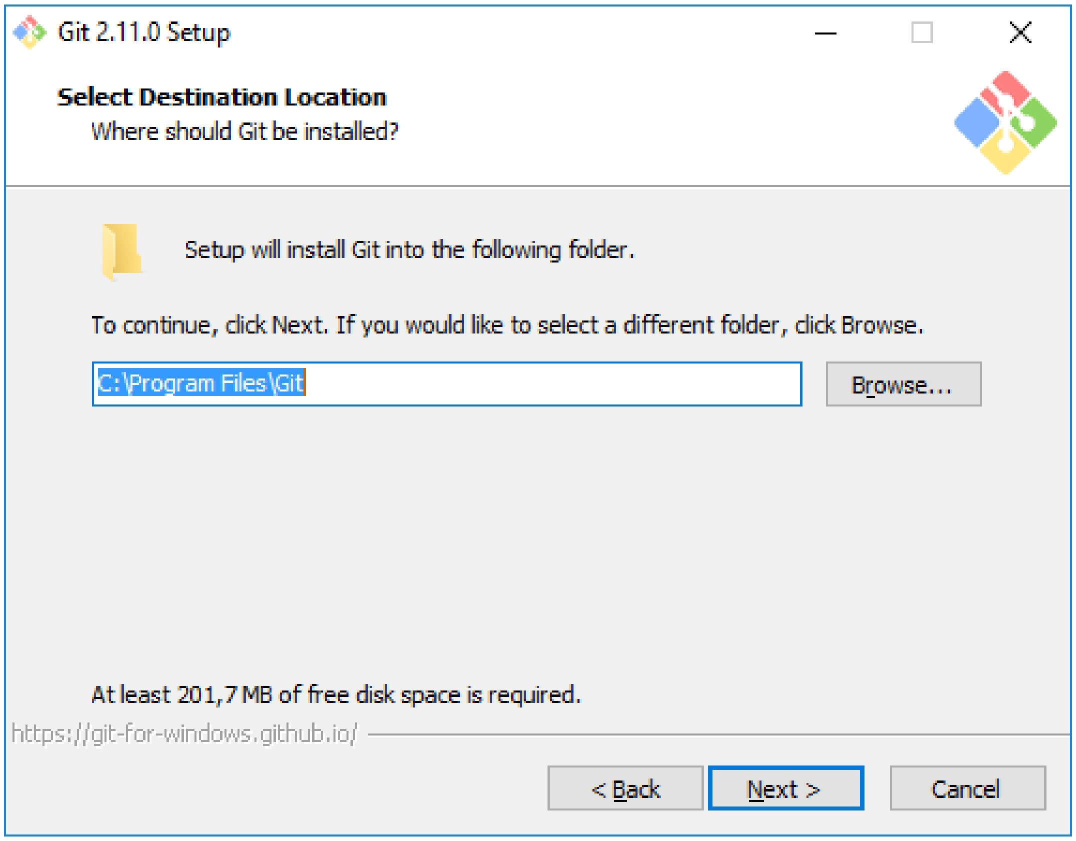
Step 1: Go to the main page of GitHub and create an account by picking a username, an email address and a password.

Step 2: Choose Unlimited public repositories for free.

Step 3: Submit to finish the sign up.

Git

Download the official build of Git from its website [51]. After visiting the previous referred link the download will start automatically. Run the executable file in order the installation process to start. The pictures below demonstrate the installation steps. It is important to follow the instructions below exactly as they are demonstrated in the pictures.


Git 2.11.0 Setup

Select Destination Location

Where should Git be installed?

Setup will install Git into the following folder.

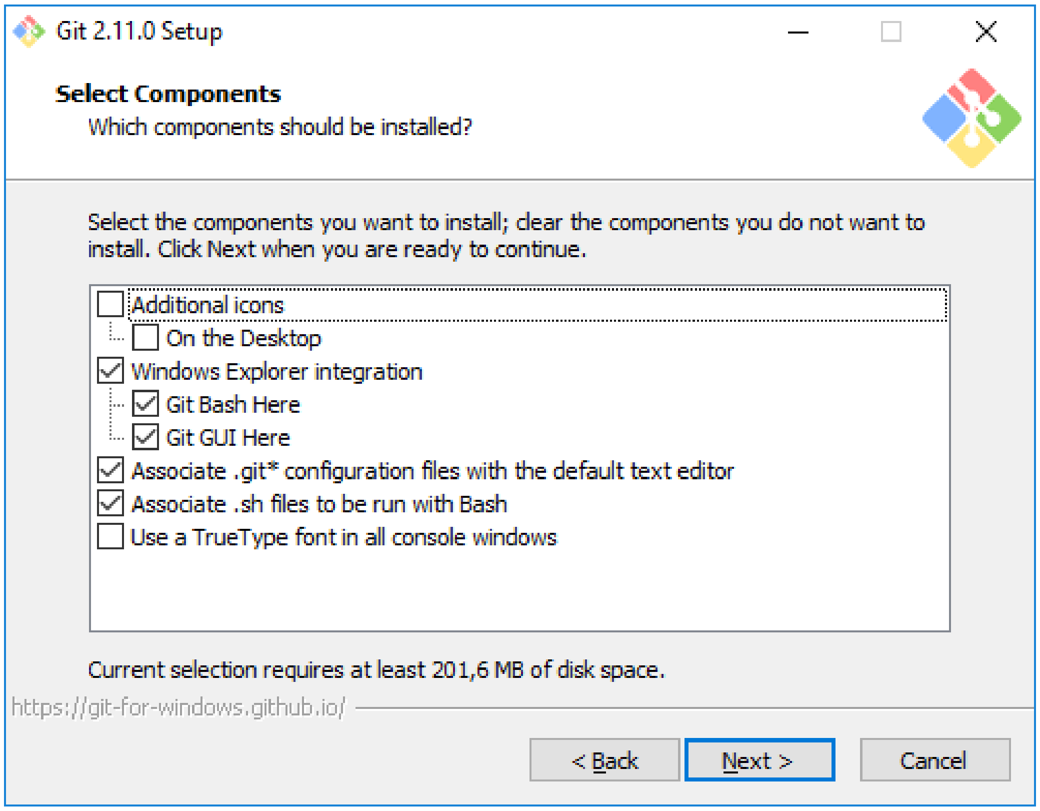
To continue, click Next. If you would like to select a different folder, click Browse.

C:\Program Files\Git

At least 201,7 MB of free disk space is required.

https://git-for-windows.github.io/

Figure 41: Git Installation Step 1


Git 2.11.0 Setup

Select Components

Which components should be installed?

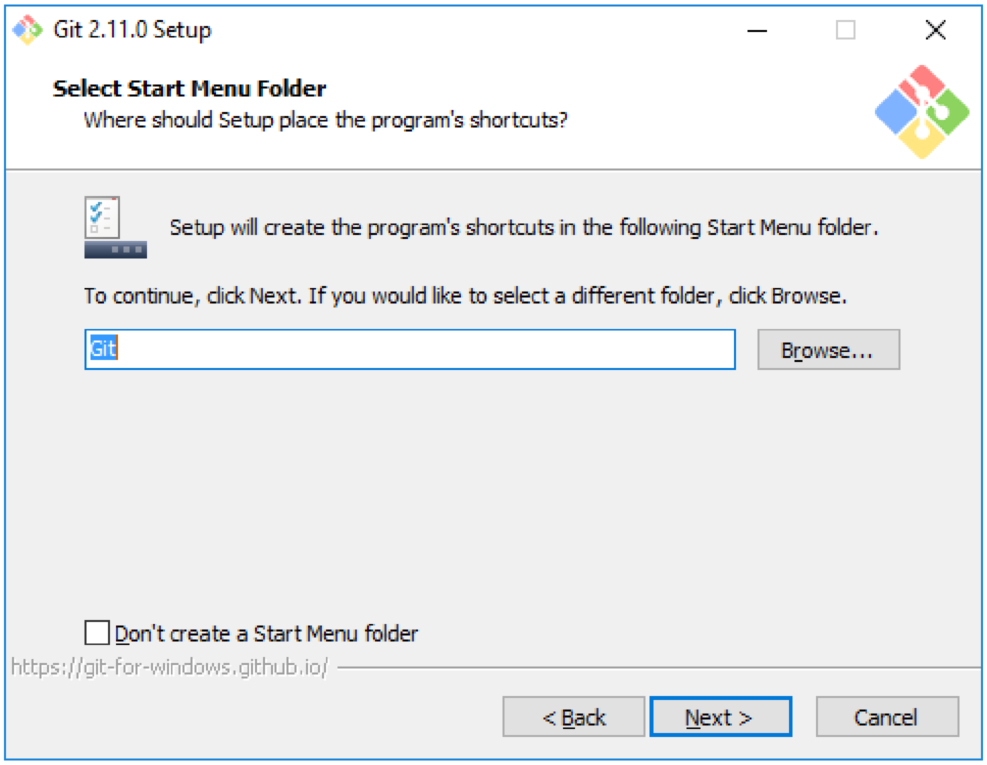
Select the components you want to install; clear the components you do not want to install. Click Next when you are ready to continue.

- [ ] Additional icons
  - [ ] On the Desktop
- [x] Windows Explorer integration
  - [x] Git Bash Here
  - [x] Git GUI Here
- [x] Associate .git* configuration files with the default text editor
- [x] Associate .sh files to be run with Bash
- [ ] Use a TrueType font in all console windows

Current selection requires at least 201,6 MB of disk space.

https://git-for-windows.github.io/

Figure 42: Git Installation Step 2



Git 2.11.0 Setup

Select Start Menu Folder

Where should Setup place the program's shortcuts?

Setup will create the program's shortcuts in the following Start Menu folder.

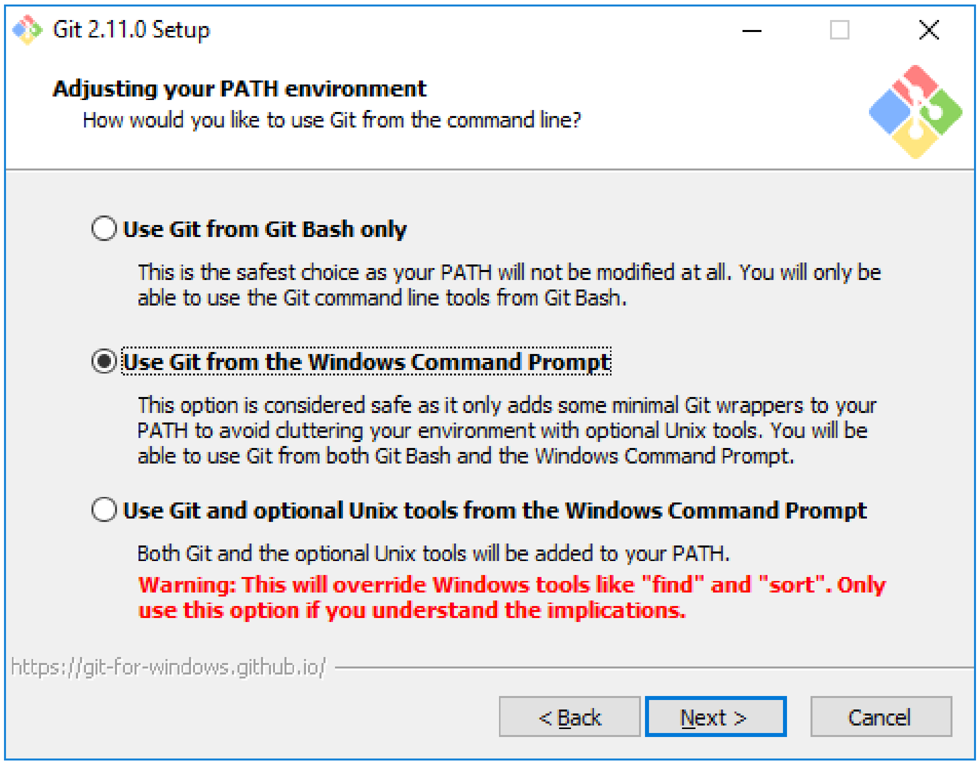
To continue, click Next. If you would like to select a different folder, click Browse.

Git

☐ Don't create a Start Menu folder

https://git-for-windows.github.io/

Figure 43: Git Installation Step 3



Git 2.11.0 Setup

Adjusting your PATH environment

How would you like to use Git from the command line?

○ **Use Git from Git Bash only**

This is the safest choice as your PATH will not be modified at all. You will only be able to use the Git command line tools from Git Bash.

● **Use Git from the Windows Command Prompt**

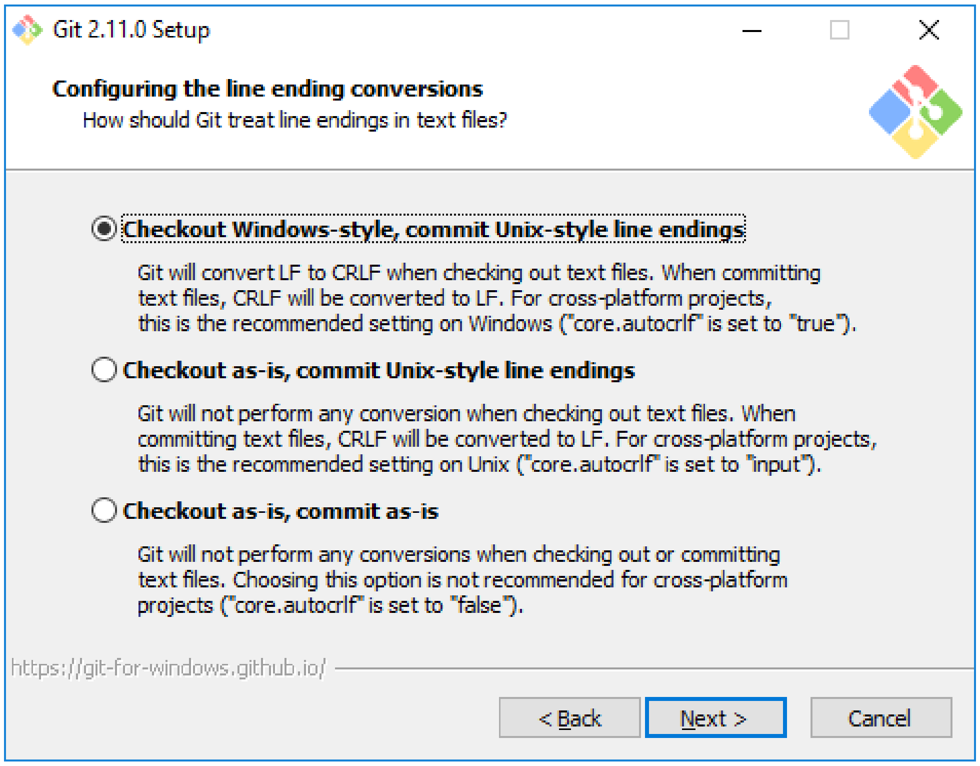
This option is considered safe as it only adds some minimal Git wrappers to your PATH to avoid cluttering your environment with optional Unix tools. You will be able to use Git from both Git Bash and the Windows Command Prompt.

○ **Use Git and optional Unix tools from the Windows Command Prompt**

Both Git and the optional Unix tools will be added to your PATH.
Warning: This will override Windows tools like "find" and "sort". Only use this option if you understand the implications.

https://git-for-windows.github.io/

Figure 44: Git Installation Step 4



Git 2.11.0 Setup

Configuring the line ending conversions

How should Git treat line endings in text files?

[✓ **Checkout Windows-style, commit Unix-style line endings**]

Git will convert LF to CRLF when checking out text files. When committing text files, CRLF will be converted to LF. For cross-platform projects, this is the recommended setting on Windows ("core.autocrlf" is set to "true").

Checkout as-is, commit Unix-style line endings

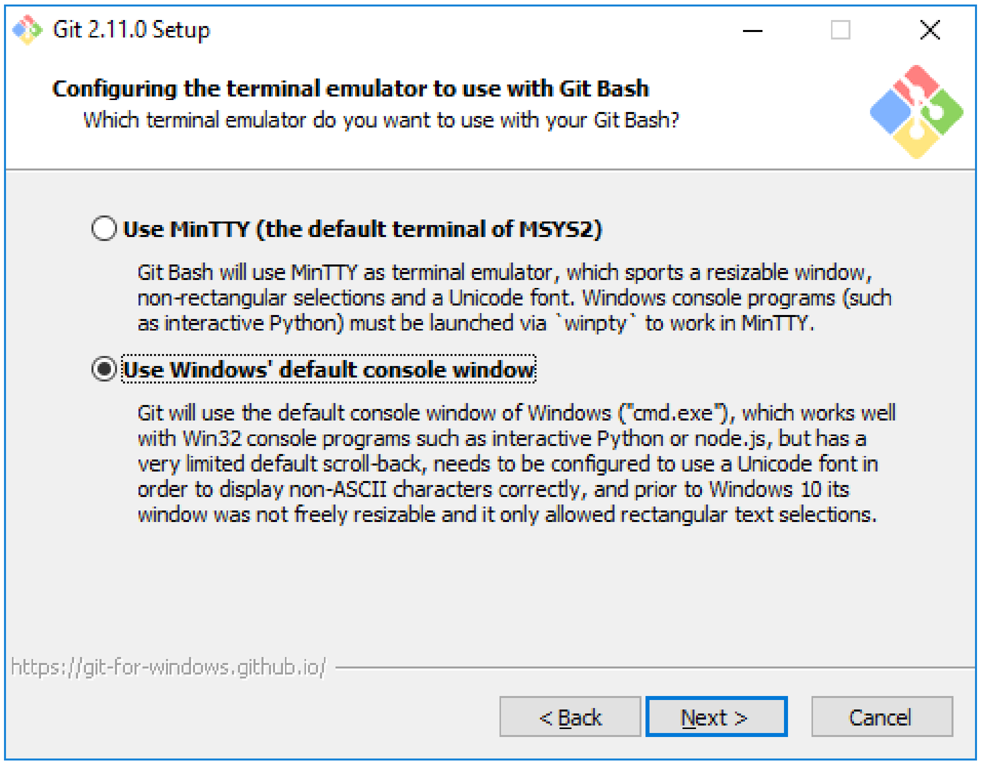
Git will not perform any conversion when checking out text files. When committing text files, CRLF will be converted to LF. For cross-platform projects, this is the recommended setting on Unix ("core.autocrlf" is set to "input").

Checkout as-is, commit as-is

Git will not perform any conversions when checking out or committing text files. Choosing this option is not recommended for cross-platform projects ("core.autocrlf" is set to "false").

https://git-for-windows.github.io/

Figure 45: Git Installation Step 4



Git 2.11.0 Setup

Configuring the terminal emulator to use with Git Bash

Which terminal emulator do you want to use with your Git Bash?

Use MinTTY (the default terminal of MSYS2)

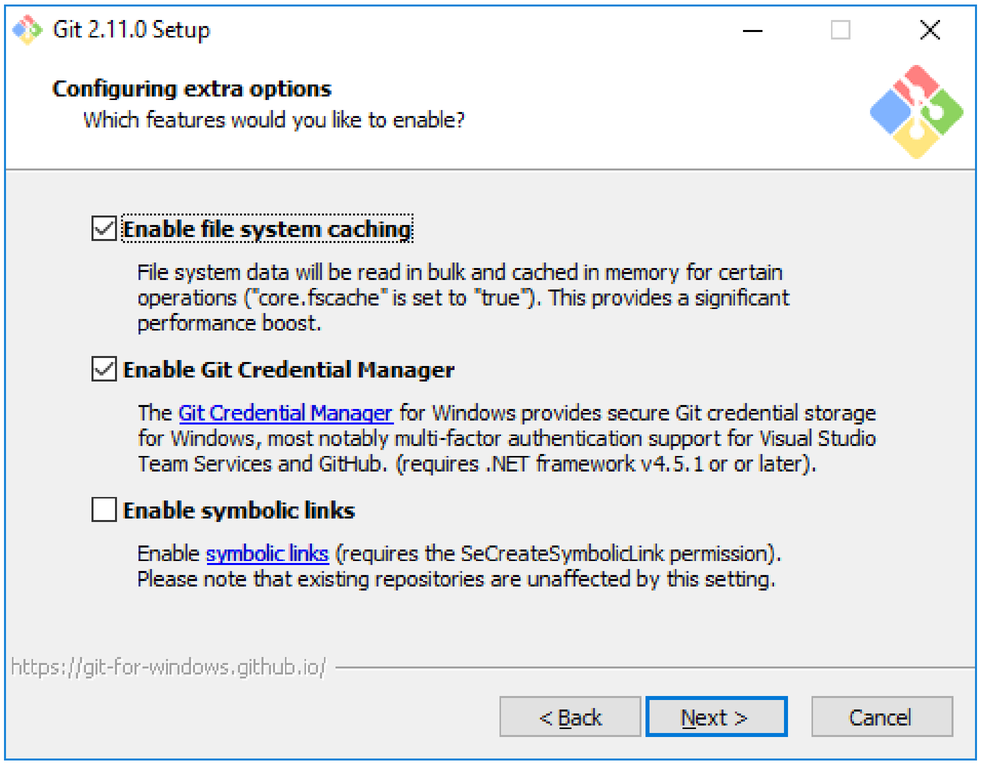
Git Bash will use MinTTY as terminal emulator, which sports a resizable window, non-rectangular selections and a Unicode font. Windows console programs (such as interactive Python) must be launched via `winpty` to work in MinTTY.

[✓ **Use Windows' default console window**]

Git will use the default console window of Windows ("cmd.exe"), which works well with Win32 console programs such as interactive Python or node.js, but has a very limited default scroll-back, needs to be configured to use a Unicode font in order to display non-ASCII characters correctly, and prior to Windows 10 its window was not freely resizable and it only allowed rectangular text selections.

https://git-for-windows.github.io/

Figure 46: Git Installation Step 4


Git 2.11.0 Setup

Configuring extra options

Which features would you like to enable?

[✓] **Enable file system caching**

File system data will be read in bulk and cached in memory for certain operations ("core.fscache" is set to "true"). This provides a significant performance boost.

[✓] **Enable Git Credential Manager**

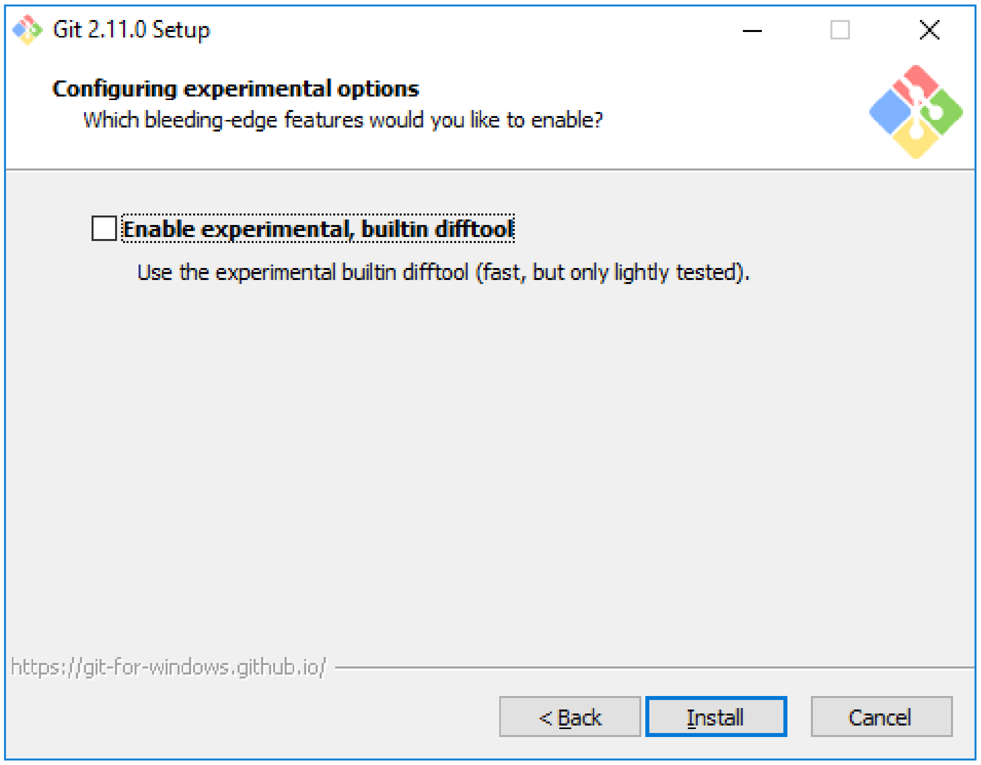
The Git Credential Manager for Windows provides secure Git credential storage for Windows, most notably multi-factor authentication support for Visual Studio Team Services and GitHub. (requires .NET framework v4.5.1 or or later).

[ ] **Enable symbolic links**

Enable symbolic links (requires the SeCreateSymbolicLink permission). Please note that existing repositories are unaffected by this setting.

https://git-for-windows.github.io/

Figure 47: Git Installation Step 5


Git 2.11.0 Setup

Configuring experimental options

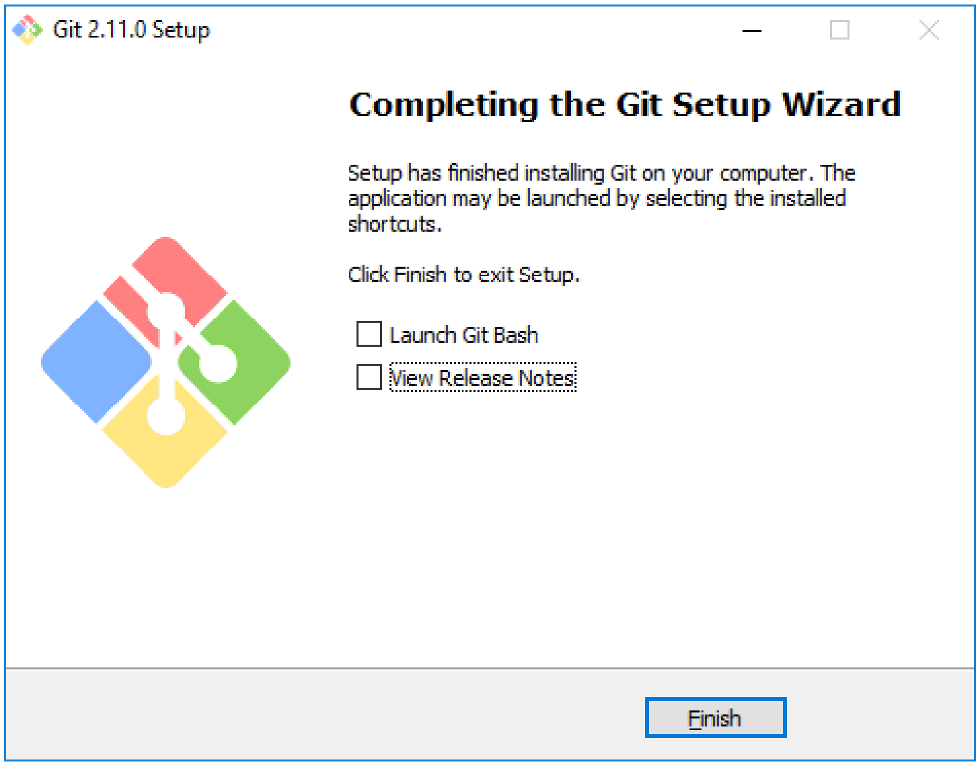
Which bleeding-edge features would you like to enable?

[ ] **Enable experimental, builtin difftool**

Use the experimental builtin difftool (fast, but only lightly tested).

https://git-for-windows.github.io/

Figure 48: Git Installation Step 6



Git 2.11.0 Setup

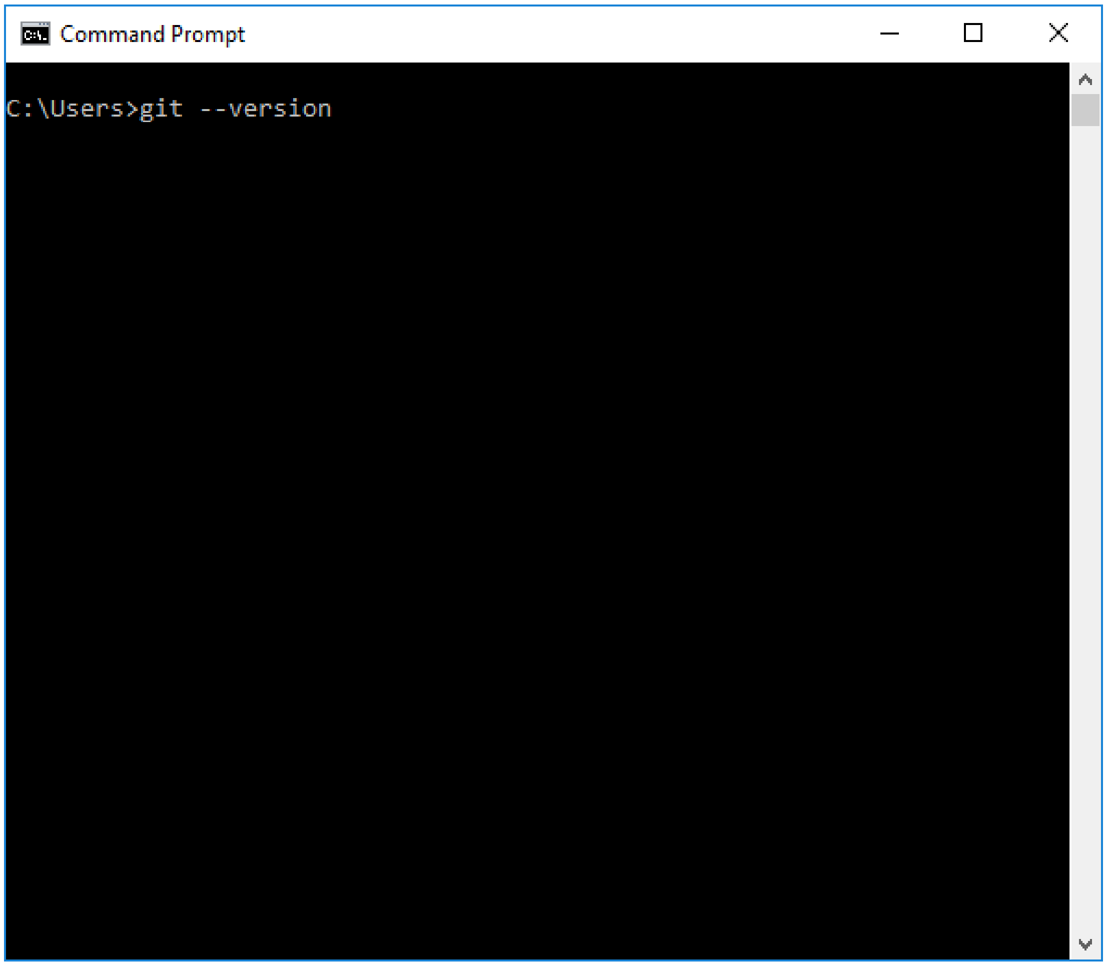
Completing the Git Setup Wizard

Setup has finished installing Git on your computer. The application may be launched by selecting the installed shortcuts.

Click Finish to exit Setup.

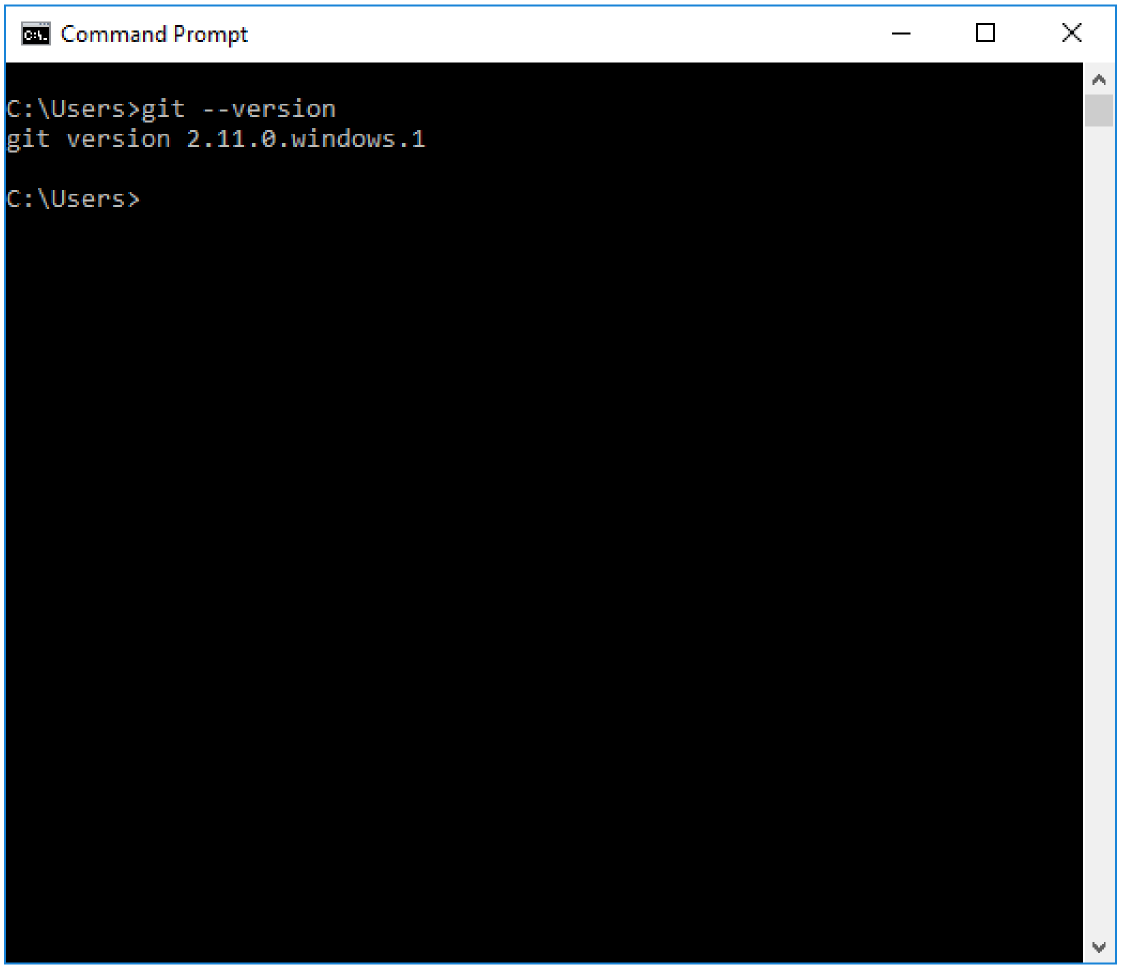
Figure 49: Git Installation Step 7

Now the installation process has finished and Git is ready to be used. There is no need of system environment configuration since this has been done by the previous installation. To verify that Git was installed successfully on the personal computer, try the process below using a windows command window.



```
C:\Users>git --version
```

Figure 50: Verification of Git installation



```
C:\Users>git --version
git version 2.11.0.windows.1

C:\Users>
```

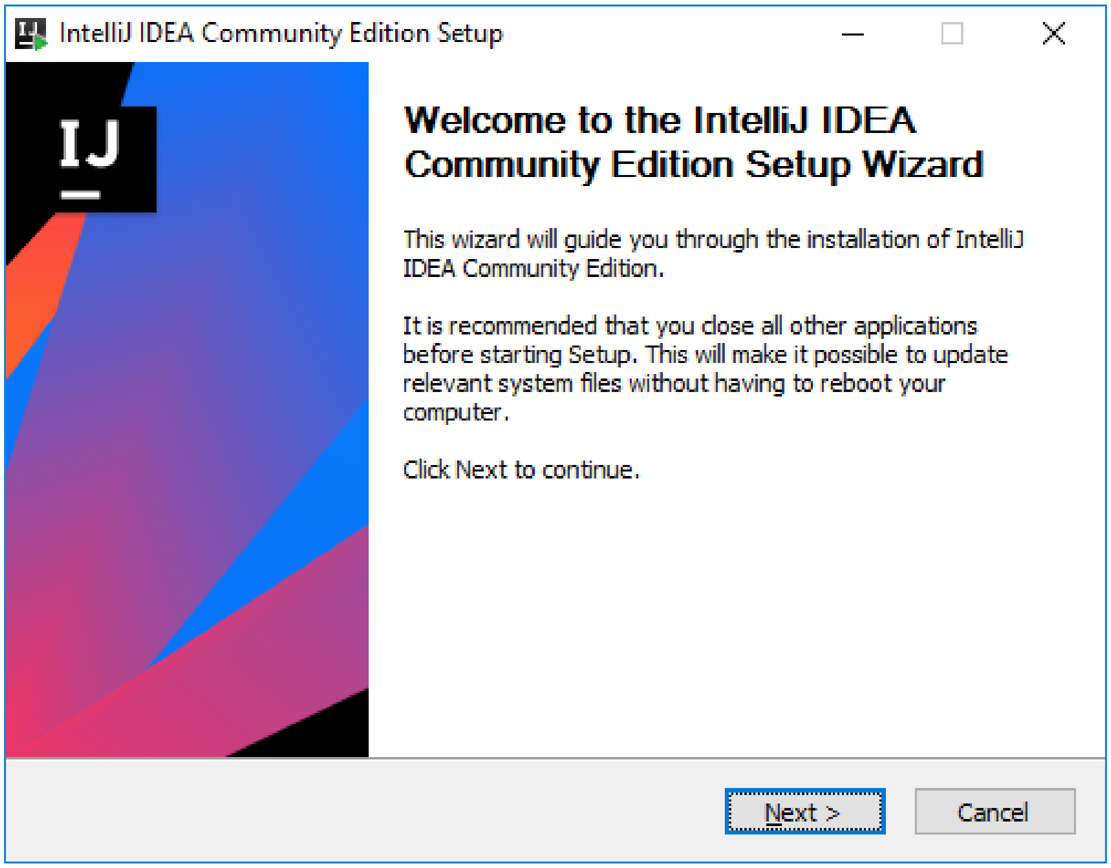
Figure 51: Verification of Git installation

As the above pictures demonstrate, the verification process includes two parts. First open a cmd console, type "git --version" and press enter. The message that should be displayed is the version of git that was installed.

IntelliJ

After installing Git the next step is to install IntelliJ IDE. This section contains the appropriate information to install, setup and configure the IntelliJ IDE as well as integrate the project.

Download the Windows version [53] of IntelliJ from the official site of JetBrains. Run the executable file as administrator and follow the instruction exactly as they are demonstrated by the below pictures.



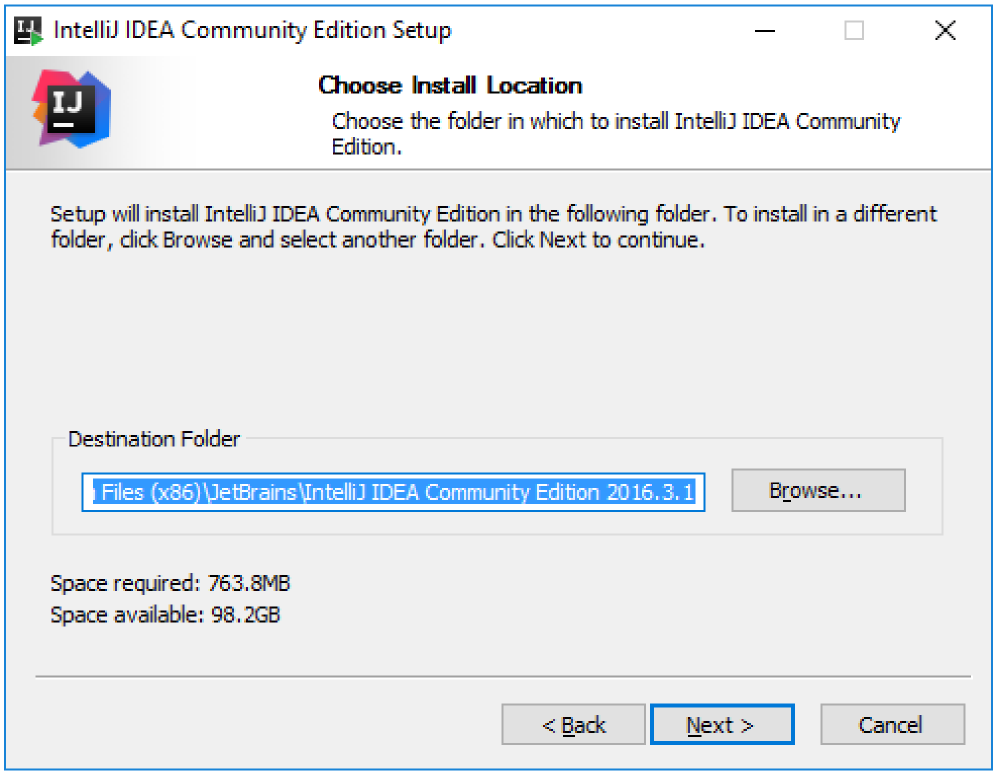
Welcome to the IntelliJ IDEA Community Edition Setup Wizard

This wizard will guide you through the installation of IntelliJ IDEA Community Edition.

It is recommended that you close all other applications before starting Setup. This will make it possible to update relevant system files without having to reboot your computer.

Click Next to continue.

Figure 52: IntelliJ Installation Step 1


IntelliJ IDEA Community Edition Setup

Choose Install Location

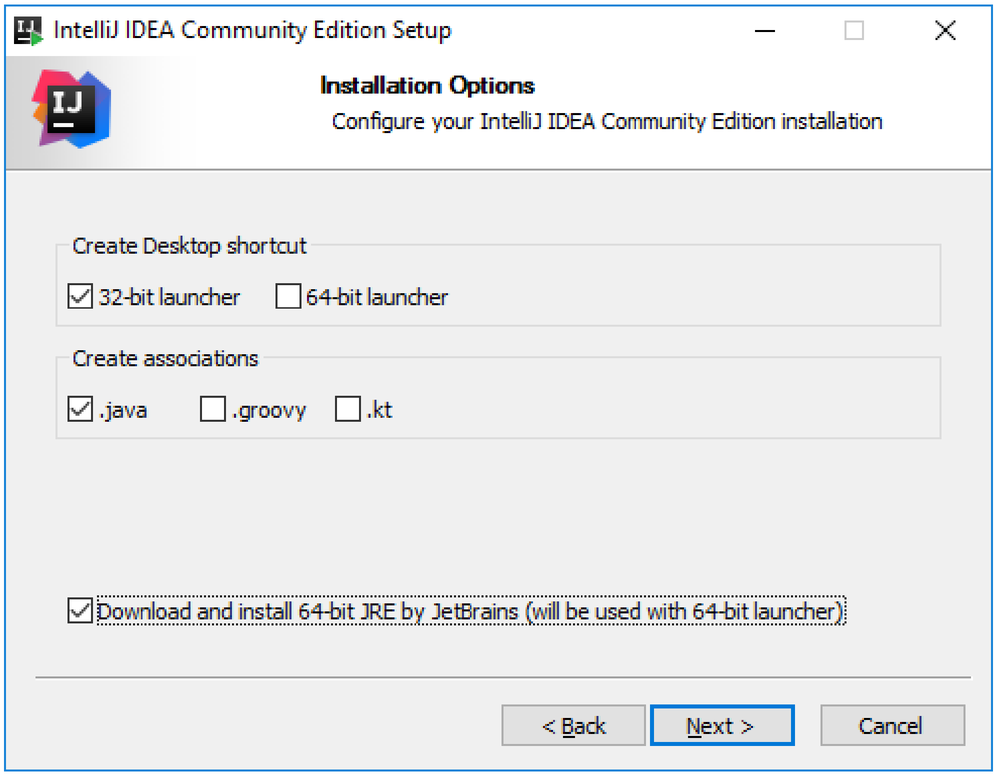
Choose the folder in which to install IntelliJ IDEA Community Edition.

Setup will install IntelliJ IDEA Community Edition in the following folder. To install in a different folder, click Browse and select another folder. Click Next to continue.

Destination Folder: Files (x86)\JetBrains\IntelliJ IDEA Community Edition 2016.3.1

Space required: 763.8MB
Space available: 98.2GB

Figure 53: IntelliJ Installation Step 2


IntelliJ IDEA Community Edition Setup

Installation Options

Configure your IntelliJ IDEA Community Edition installation

Create Desktop shortcut

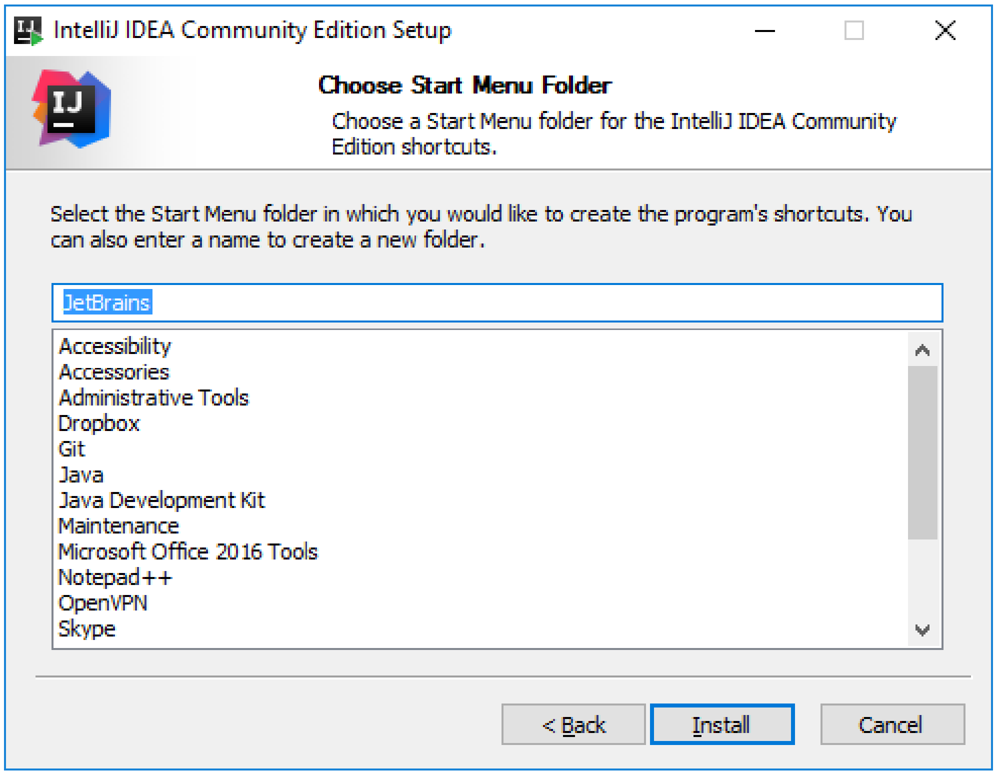
[✓] 32-bit launcher [ ] 64-bit launcher

Create associations

[✓] .java [ ] .groovy [ ] .kt

[✓] Download and install 64-bit JRE by JetBrains (will be used with 64-bit launcher)

Figure 54: IntelliJ Installation Step 3


IntelliJ IDEA Community Edition Setup

Choose Start Menu Folder

Choose a Start Menu folder for the IntelliJ IDEA Community Edition shortcuts.

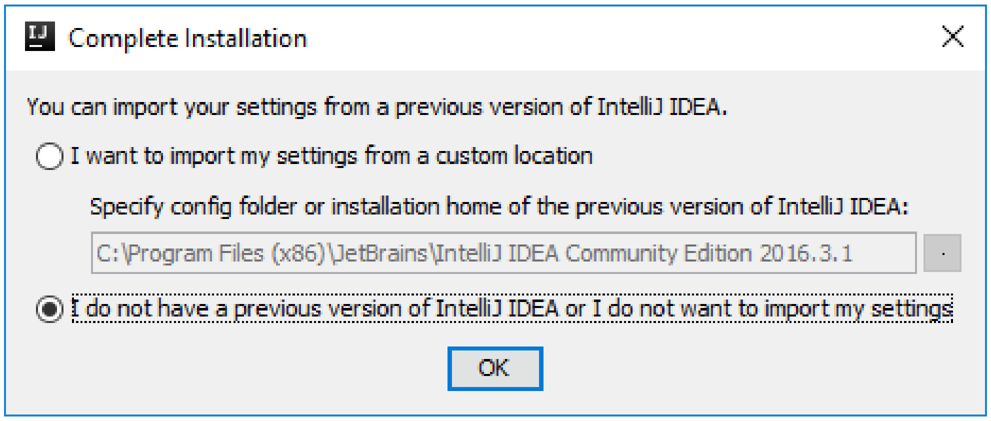
Select the Start Menu folder in which you would like to create the program's shortcuts. You can also enter a name to create a new folder.

JetBrains

Accessibility
Accessories
Administrative Tools
Dropbox
Git
Java
Java Development Kit
Maintenance
Microsoft Office 2016 Tools
Notepad++
OpenVPN
Skype

Figure 55: IntelliJ Installation Step 4

After the successful installation a shortcut on the desktop will appear. Double click on the shortcut to open the IDE. Then follow the next steps that are demonstrated by the below pictures to configure the IntelliJ IDE. The below steps should be followed exactly for the appropriate configuration setup.


Complete Installation

You can import your settings from a previous version of IntelliJ IDEA.

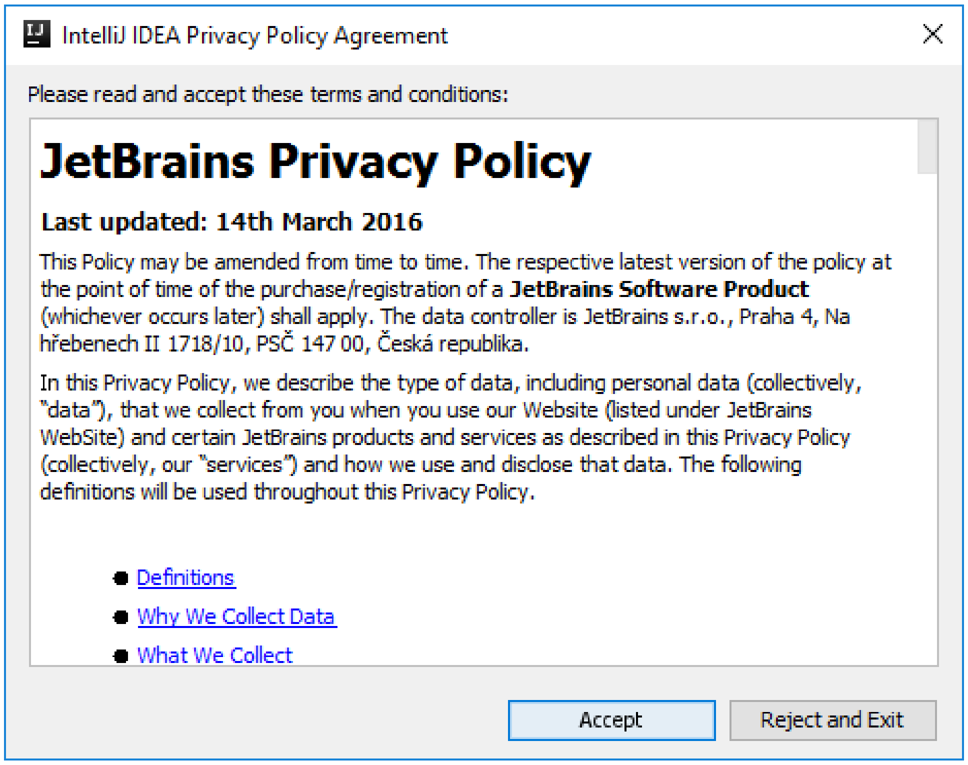
○ I want to import my settings from a custom location

Specify config folder or installation home of the previous version of IntelliJ IDEA:

C:\Program Files (x86)\JetBrains\IntelliJ IDEA Community Edition 2016.3.1

[✓ I do not have a previous version of IntelliJ IDEA or I do not want to import my settings]

Figure 56: IntelliJ Configuration Step 1


IntelliJ IDEA Privacy Policy Agreement

Please read and accept these terms and conditions:

JetBrains Privacy Policy

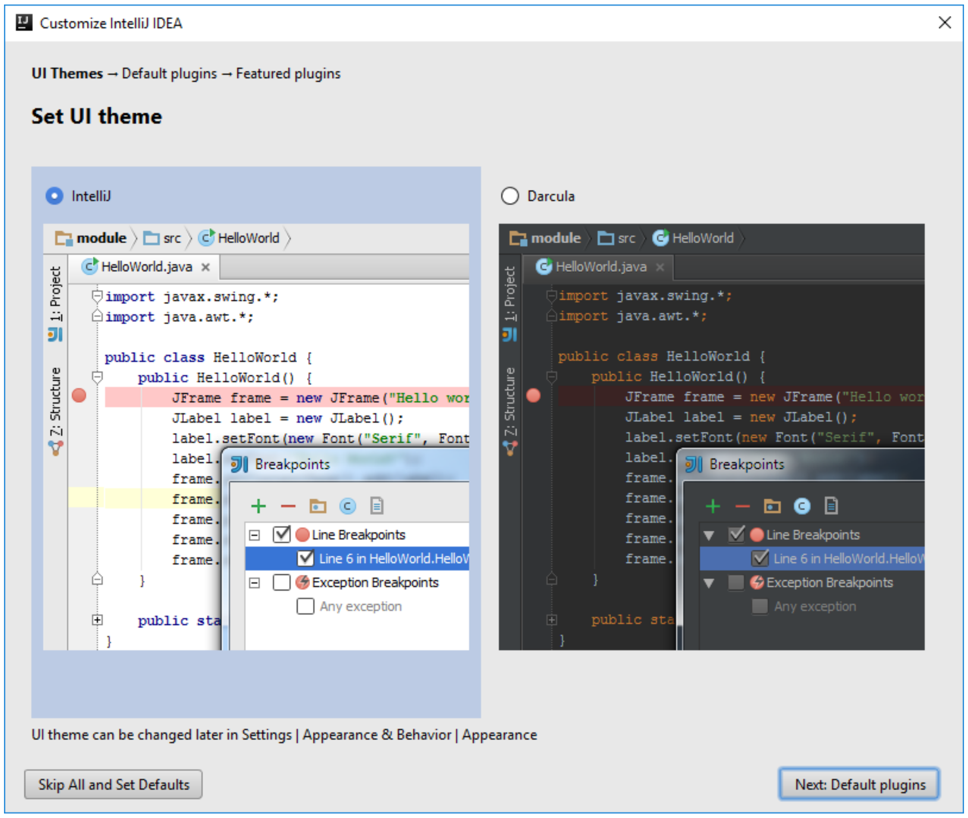
Last updated: 14th March 2016

This Policy may be amended from time to time. The respective latest version of the policy at the point of time of the purchase/registration of a **JetBrains Software Product** (whichever occurs later) shall apply. The data controller is JetBrains s.r.o., Praha 4, Na hřebenech II 1718/10, PSČ 147 00, Česká republika.

In this Privacy Policy, we describe the type of data, including personal data (collectively, "data"), that we collect from you when you use our Website (listed under JetBrains WebSite) and certain JetBrains products and services as described in this Privacy Policy (collectively, our "services") and how we use and disclose that data. The following definitions will be used throughout this Privacy Policy.

- Definitions
- Why We Collect Data
- What We Collect

Figure 57: IntelliJ Configuration Step 2



Customize IntelliJ IDEA

UI Themes → Default plugins → Featured plugins

Set UI theme

(●) IntelliJ ( ) Darcula

module › src › HelloWorld

HelloWorld.java

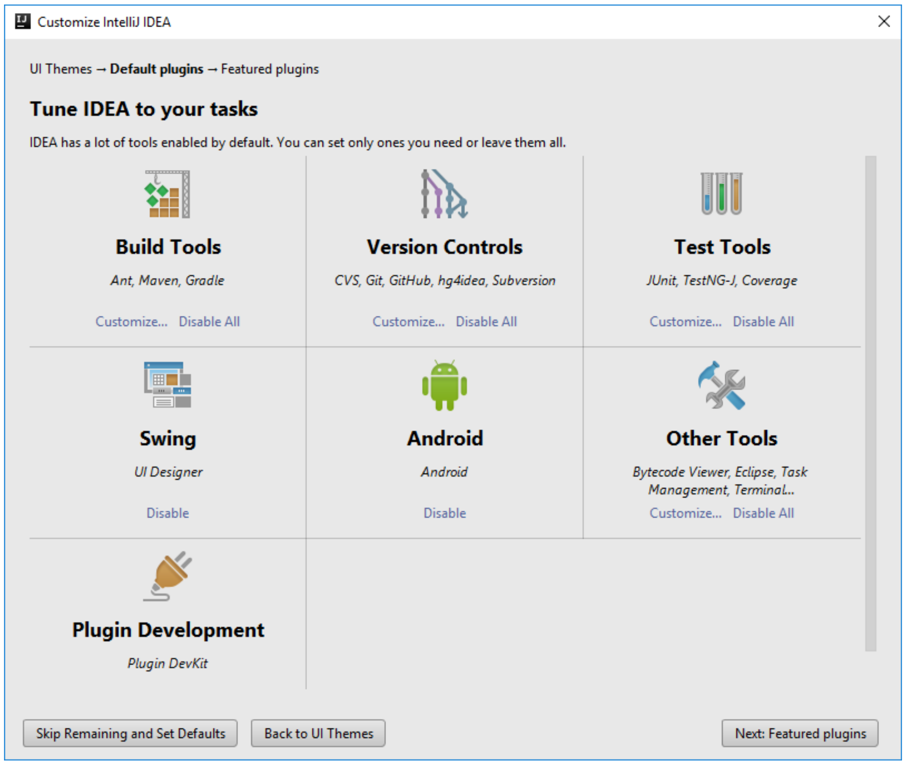
```
import javax.swing.*;
import java.awt.*;

public class HelloWorld {
    public HelloWorld() {
        JFrame frame = new JFrame("Hello wor
        JLabel label = new JLabel();
        label.setFont(new Font("Serif", Font
        label.
        frame.
        frame.
        frame.
        frame.
        frame.
    }

    public sta
}
```

Breakpoints

- [✓] Line Breakpoints
  - [✓] Line 6 in HelloWorld.HelloV
- [ ] Exception Breakpoints
  - [ ] Any exception

UI theme can be changed later in Settings | Appearance & Behavior | Appearance

Figure 58: IntelliJ Configuration Step 3



Customize IntelliJ IDEA

UI Themes → **Default plugins** → Featured plugins

Tune IDEA to your tasks

IDEA has a lot of tools enabled by default. You can set only ones you need or leave them all.

Build Tools
Ant, Maven, Gradle
Customize... Disable All

Version Controls
CVS, Git, GitHub, hg4idea, Subversion
Customize... Disable All

Test Tools
JUnit, TestNG-J, Coverage
Customize... Disable All

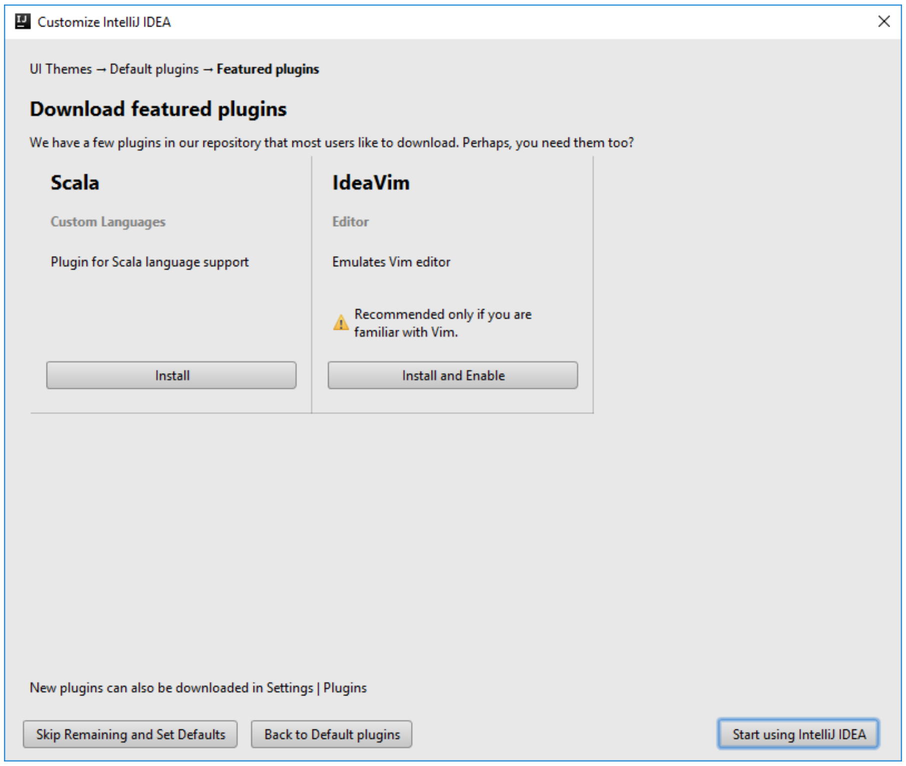
Swing
UI Designer
Disable

Android
Android
Disable

Other Tools
Bytecode Viewer, Eclipse, Task Management, Terminal...
Customize... Disable All

Plugin Development
Plugin DevKit

Figure 59: IntelliJ Configuration Step 4



Customize IntelliJ IDEA

UI Themes → Default plugins → **Featured plugins**

Download featured plugins

We have a few plugins in our repository that most users like to download. Perhaps, you need them too?

Scala

Custom Languages

Plugin for Scala language support

IdeaVim

Editor

Emulates Vim editor

Recommended only if you are familiar with Vim.

New plugins can also be downloaded in Settings | Plugins

Figure 60: IntelliJ Configuration Step 5

After clicking the "Start using IntelliJ IDEA" button a new window will appear. This window should be closed. Now from the following window click "Check out from Version Control" and select GitHub from the drop down menu.

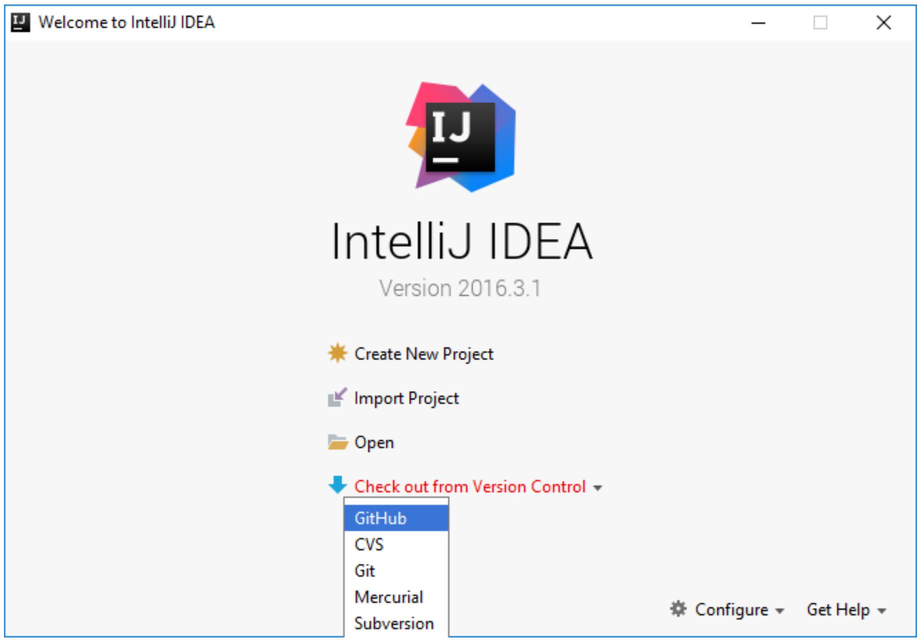
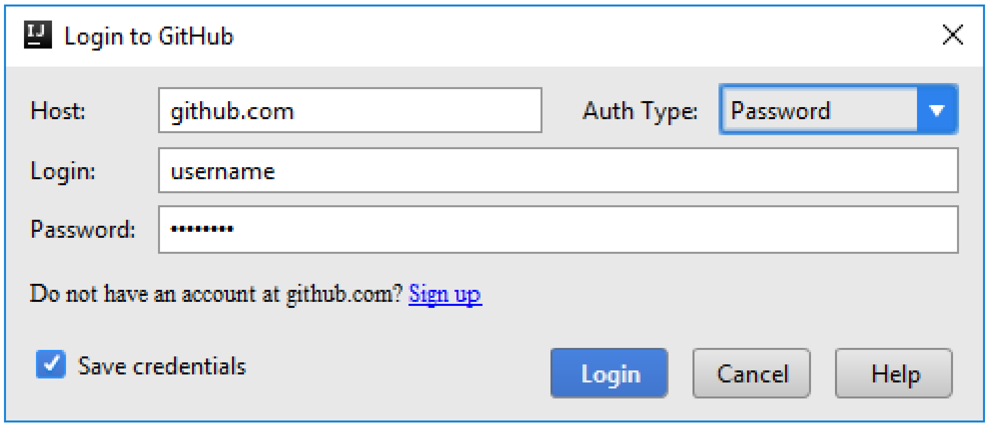


Figure 61: IntelliJ Configuration Step 6

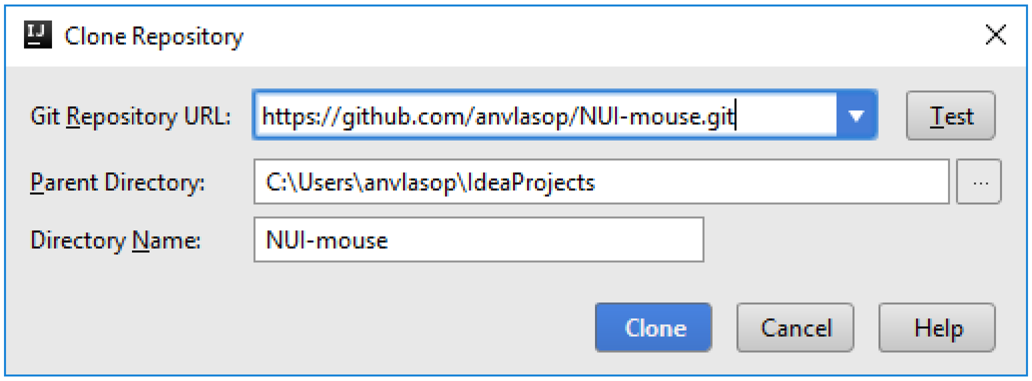
The next picture shows the login process for GitHub. Here the engineer should login using the credentials (username and password) that were created on the GitHub web page. On the "Auth Type" drop down menu the engineer should select "Password" and then enter his credentials.



Host: github.com
Auth Type: Password
Login: username
Password: ••••••••
[✓] Save credentials

Figure 62: IntelliJ Configuration Step 7

After the successful login process the next step is to enter the Git Repository URL as shown in the picture below. By clicking the “Clone” button the IDE will clone the remote repository into the local defined folder.

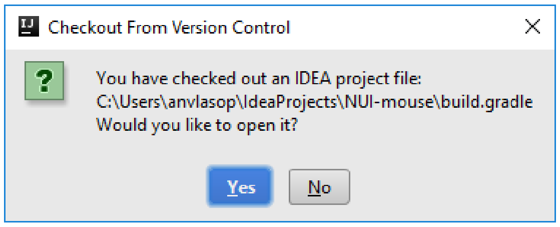

Clone Repository

Git Repository URL: https://github.com/anvlasop/NUI-mouse.git

Parent Directory: C:\Users\anvlasop\IdeaProjects

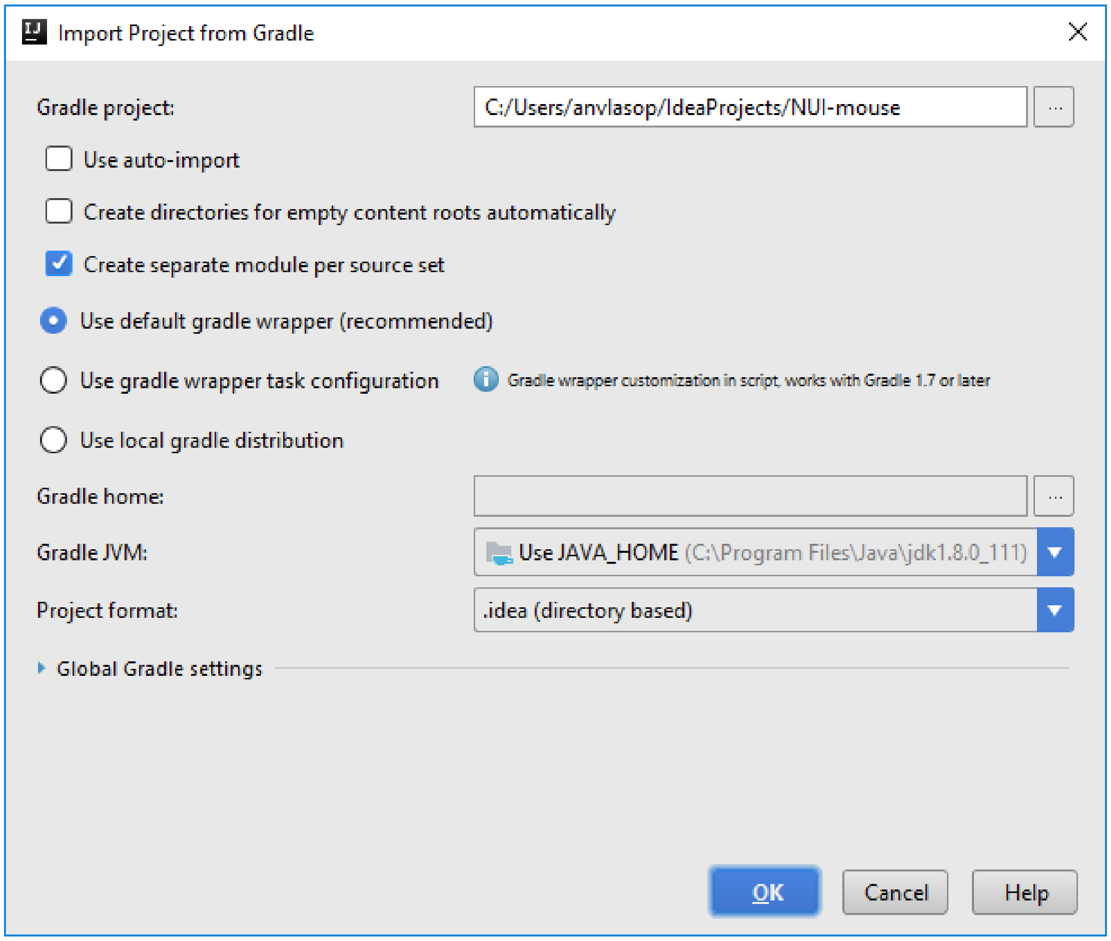
Directory Name: NUI-mouse

Figure 63: IntelliJ Configuration Step 8


Checkout From Version Control

You have checked out an IDEA project file:
C:\Users\anvlasop\IdeaProjects\NUI-mouse\build.gradle
Would you like to open it?

Figure 64: IntelliJ Configuration Step 9


Import Project from Gradle

Gradle project: C:/Users/anvlasop/IdeaProjects/NUI-mouse

- [ ] Use auto-import
- [ ] Create directories for empty content roots automatically
- [x] Create separate module per source set

● Use default gradle wrapper (recommended)

○ Use gradle wrapper task configuration — Gradle wrapper customization in script, works with Gradle 1.7 or later

○ Use local gradle distribution

Gradle home:

Gradle JVM: Use JAVA_HOME (C:\Program Files\Java\jdk1.8.0_111)

Project format: .idea (directory based)

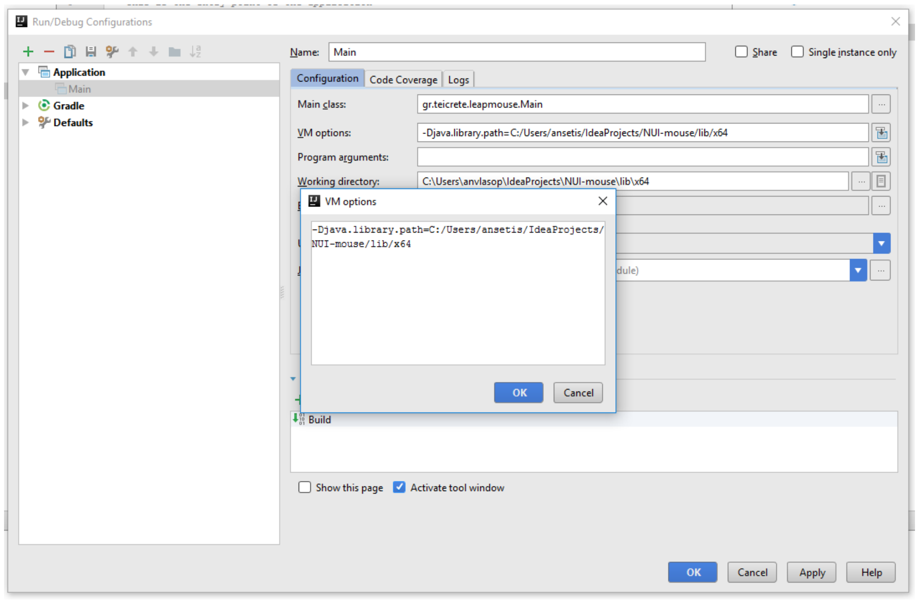
Figure 65: IntelliJ Configuration Step 10

Step 12: After the above step the project will open by the IDE. The next step is to define the SDK of the project. To do that go to the class Main and click on the “Setup SDK” link that is shown on the top of the class’ tab. On the new window that will appear click the “Configure” button. Next click on the “+” icon on the upper left corner of the new opened window and select the folder where the Java JDK is installed. It should be at this path: C:/Program Files/Java/jdk<version>

Step 13: Now go to View->Tool Windows->Gradle. The Gradle pane will appear on the right side of the IDE. To build the project, double click on the “build” task which can be found from the previous mentioned pane under leap-mouse->Tasks->build. The build should be successful.

Step 14: The final step of IntelliJ IDE configuration process is to set the path to the native Leap Motion libraries. For this purpose follow the next sub-steps:

1. Select the Run > Edit Configurations… menu command.
2. Set the VM Options field to set the -Djava.library.path parameter to the path to the proper folder containing the Leap Motion native libraries as shown in the picture below.
3. Click ok



Name: Main
[ ] Share [ ] Single instance only
Main class: gr.teicrete.leapmouse.Main
VM options: -Djava.library.path=C:/Users/ansetis/IdeaProjects/NUI-mouse/lib/x64
Program arguments:
Working directory: C:\Users\anvlasop\IdeaProjects\NUI-mouse\lib\x64

```
-Djava.library.path=C:/Users/ansetis/IdeaProjects/
NUI-mouse/lib/x64
```

[ ] Show this page [✓] Activate tool window

Figure 66: IntelliJ Installation Step 14

Now the project is ready to run. Go to the class Main and run the application. Be sure that the Leap Motion controller is connected.