

CyprusRun Report



Made by
Arda Tatar

Table Of Contents



Table of Contents

1. Introduction
2. Team Member
3. Game Concept
4. Gameplay and Features
5. History Section
6. Development Tools and Workflow
7. Challenges and Solutions
8. Testing and Feedback
9. Future Improvements
10. Conclusion
11. Screenshots
12. References
- 2-1 Page Setup
- 2-2 Overview

1.Introduction



The word "engineer" comes from the Latin word "ingenium", which means being talented, smart, and skilled. A good engineer should have these qualities in order to do well in their job and life. But having talent alone is not enough to become an engineer. A person also needs to complete an engineering program at a good school or university.

At the end of their studies, and before graduation, engineering students must show what they have learned by doing a project. This project proves that they can use their knowledge and finish something important. The size and difficulty of the project are planned so that students can complete it in the time they are given.

1.2 What is the Aim of an Internship Project?

The purpose of an internship project becomes clear to most students after they complete it and present their work.

By the time you reach this stage, you will have learned many useful things that will help you in real life, especially when you begin working as an engineer.

Here are some of the things you can learn during an internship project:

- Knowledge and Expertise: learning more about a topic and gaining specialized skills.
- Time Punctuality: working on a project within a set time frame.
- Time Management: planning your tasks to finish the project on time.

- Research: collecting information from different sources related to your project topic.
- Analytical Thinking: understanding and analyzing the information you collect.

You can also improve in areas such as:

- Taking Responsibility: making decisions and recording them during your project.
- Engineering Design: creating ideas and making decisions based on your research and analysis.
- Individual Creativity: thinking of new or better solutions connected to your topic.
- English Reading & Writing Skills: your English skills may improve by reading, writing, and reporting during the project.
- Communication Skills: talking about your work with your supervisor or teammates helps you become better at technical communication.
- Self-Confidence: explaining your project at the end of your internship will increase your confidence in yourself and your work.

CyprusRun is a 3D running game where the player runs through a scripted terrain, a stylized landscape of Cyprus in our imagination. The objective of the game is to survive until the end of the sections. If you fail, restart again and again until you reach the endpoint. Each time the player dies, the game restarts and the score increases by one point. The game has features like fixed terrain that regenerates in each try, and what makes it unique is the inclusion of a history section that educates players about the rich history of Cyprus with its historical

timeline from ancient Ionians to the modern era and even the imagination of future Cyprus and the people of Cyprus. The game has a high level of reflex-based gameplay, but with this gameplay, the game gives you a rich amount of information with the history section of the game. With every try, the player learns a segment and wonders about the segment and checks what it is about.

Using This Book The main aim of this book is to help computer engineering students with the rules and steps for doing their internship projects. It shows students how to choose a project, how to write the report, how to prepare for the presentation, and how to explain and defend their work. This book answers many common questions that students may have during their internship. It can be used along with help and advice from your supervisor. The book is divided into different chapters. In Chapter One (Introduction)... In Chapter Two (General Information)...

Final Words I wish all students success in their internship projects. We also hope that during this process, students gain useful knowledge and skills in the topic they choose skills that will help them as future engineers.

2.Team Members

Name: Arda Tatar

Role: Solo Developer, Designer, Programmer



2. General Information



As a student of Software Engineering at Near East University, I had the chance to work on an internship project. This internship was not just something I had to do to complete the program, but actually a really useful experience for improving my skills and understanding how software development works in the real world.

Instead of learning everything from books or slides, I learned things directly from the field. I solved problems, developed code, tested features, and sometimes even fixed bugs that didn't seem like a problem at first. I also realized how important it is to work regularly and communicate with supervisors or teammates just like we do with friends from school when we're stuck. They gave me a lot of useful feedback, especially about my game project.

The internship gave me a real taste of what it's like to work as a software engineer. I didn't just write random code; I had to really think about what I was building, plan it properly, and sometimes even start from scratch when something didn't work as expected. It helped me grow both technically and personally.

Topics for an Internship Project

There are many possible topics you can choose from during your internship in Software Engineering. Here are a few examples:

Software Development: Probably the most common topic building real applications for desktop, web, or mobile. You can use C#, Java, Python, JavaScript, etc. Personally, I think this is where we apply most of the things we've learned.

Artificial Intelligence: If you're into smart systems, this is a cool area. You can try out machine learning models, play with data, or create something like a classifier or recommendation system.

Neural Networks / Genetic Algorithms: These are more advanced than they seem, and I won't lie they're not always easy. But they're really interesting if you're curious about how machines can "think" or evolve to solve problems.

Expert Systems: These are knowledge-based decision systems. You give them rules and logic, and they can help make decisions automatically.

Web & Internet Technologies: This includes full-stack development, frontend-backend communication, working with APIs, or cloud services. There's a lot you can do here, and most companies need developers in this field.

Multimedia & Image Processing: This is a fun one if you enjoy working with images, sound, or video. For example: face recognition, speech-to-text, or detecting objects in a video. Let's say your family member wants

someone removed from a photo if you can do that, it makes the photo even more heartwarming.

These are just examples. What really matters is choosing something you enjoy and can learn from. My advice: pick something that's inside your comfort zone at first because learning is hard, and having fun while doing it makes everything easier.

The internship project is supposed to be your own work, so as a student, you're responsible for everything inside it. We need to choose a topic, find a supervisor from the academic staff in the Software Engineering Department, and plan our time properly with their support.

At our faculty, we can't just do the internship project on our own we need a supervisor to guide us. So yeah, they assign someone to help us out during the process. Their job is to support us, but that doesn't mean they'll do everything for us. It's still up to us to stay on track, meet the deadlines, and get the work done.

One thing I really wanted to learn is respect and yeah, we got respect. You have to communicate properly with your supervisor and treat them well. They're not just here to judge what you've done; they're actually there to help you do it better. Sometimes they might not know your topic, and that's okay.

They still trying to help you. It's on you to explain them clearly what you want from them or project and ask the right questions whenever you get stuck.

At the end of the day, this is your project and you're the one in charge. The supervisor is like your co-pilot they're here to guide you, not to fly the whole thing for you.

3.Game Concept



CyprusRun is a 3D running game where the player runs through a scripted terrain, a stylized version of Cyprus in our imagination. The terrain isn't just a simple background; it's carefully designed to give a special and fun experience every time. "Scripted terrain" means that each run there is a scripted terrain, but with the difficulty of the game, these scripted terrains are feeling good for the player. Every run, you are exploring a new part of the map, so the game feels fresh and cool in a way. With the obstacles and structures in the way, it takes the player inside of it. The "stylized landscape of Cyprus" is inspired by the island's nature and history, but it's not a realistic copy. Instead, it's an artistic and creative version, with things like reimagined seaports, colorful Mediterranean plants, and Unity's cool vision that show Cyprus's culture and past in a fun way.

The main goal of the game is to survive and reach the end of each section. This is the heart of the gameplay. Players must stay focused and react quickly to avoid obstacles and dangers. The game is split into 4 sections, each one with its own look and feel. This shows different

parts of the island or different time periods of the island. Players have to finish these sections without failing. If they do fail, they start again from the beginning. This adds a challenge and a reason to keep improving and inch to see other sections. Each section also reveals more about the game's world and history as players progress.

If you fail, you restart again and again until you reach the goal. This is a big part of the game's design. It's okay to fail, because each try helps you get better. The game encourages you to keep playing, learn from mistakes, and get used to the patterns. The idea is that every run makes you a little better and brings you closer to success. This retry system also supports the educational side of the game. Each attempt gives players a chance to learn something new.

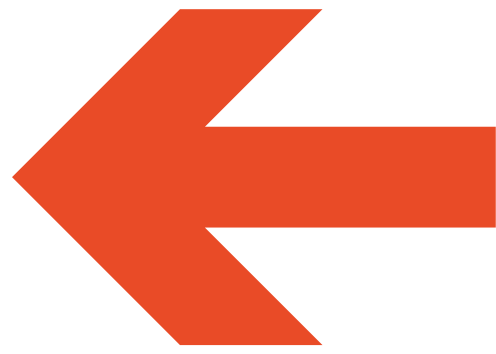
Each time you die, the game restarts, and your score goes up by one. The score system is simple but meaningful. It doesn't show how far you got; it shows how many times you tried. It rewards your encouragement to finish, not just success. Every time you fail, you earn a point. This turns failure into something valuable, showing your effort and how many times you've tried to improve and learn.

The game includes a fixed terrain that regenerates in every run, and its most unique feature is the history section. This section teaches players about Cyprus's long and rich history, from ancient times to today and even into a future version of the island. The "fixed terrain" means that the world layout is the same, but obstacles and small elements change

each time. This keeps the game fun and makes sure players can't just memorize a single path. This changing environment matches well with the history part of the game. The history section is not just extra; it's a big part of the game. It gives players a full timeline of Cyprus, sharing knowledge about the island's culture, people, and important moments in time. It also includes a creative look into what the future of Cyprus might look like. The goal is to teach players meanwhile they are having fun.

The game is focused on reflex-based gameplay, meaning players have to act fast, jump, slide, or dodge obstacles with perfect timing. The fast speed and changing terrain make the game exciting and hard. But the real magic is how this fast gameplay connects to learning. The history section shares useful knowledge without slowing down the game. The way the game restarts after failure is used to show history in small parts. This keeps it light and easy to understand.

4. Gameplay and Features



Fixed 3D Terrain: From my past experience, this game type is best to play with 3D terrain, so we used 3D graphics for the game.

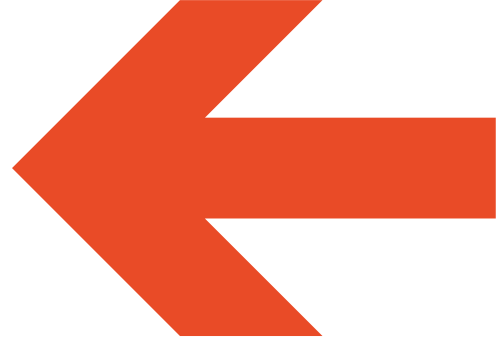
Movement-based character control: This is the main function of the game and innovative in a way that gives a totally different way of playing than Subway Surfers and Temple Run.

Run amount changes with each run attempt: Every run, the game counts your run amount so you know your run amount, and it adds some competition to the game.

In-game UI with access to a history section: The game has an in-game UI that helps you to reach the history section.

Everytime the player restarts, they see a new part of Cyprus's history. This is the smart part of the game's learning system. After a difficult try, players get a short piece of historical information. This makes them curious, and they might want to learn more about it. They can then visit the history section to read more details of the history of Cyprus. This system turns failure into a chance to learn something new. The more players play, they see the history section after a certain point. Because of failing, they will have connection to the section and consider looking at the section's history.

5. History Section



When you click the History button, six parts open up. Each one shows a different time in Cyprus. It's not too detailed, just enough for people to understand how things changed overtime.

1. Ancient Cyprus

In this phase, nationalities like the Ionians, Achaeans and others arrived. Small trade cities started growing and became rich. But after a while, the Greek kingdom invaded the island. The Ionians tried to rebel and take control of their country, but the Greeks won. Even though they won, they couldn't reduce Cyprus's autonomy.

Later, Alexander the Great came and took over Cyprus. This time it was different. Cypriots became generals in his army and even had a big role in the government. But when Alexander died, Persians took control.

Their rule was also different. They vassalised Cyprus, but Cyprus still had its own government and made its own decisions. Persians saw it as a vassal, but it was more like semi-independent.

After that, the Roman kingdom formed and took over. Cyprus became the Roman Cyprus Kingdom. They ruled for some time until the Roman Empire split. After that, Cyprus became part of the Eastern Roman Empire, also known as Byzantium.

2. Medieval and Byzantine Period

After the ancient era, Cyprus was controlled by the Byzantines for years.

They ruled the west and continued as the Roman Empire in a new form. Eventually, the empire couldn't stay together. It split again, and Cyprus stayed with the Byzantines. After the fall of the empire, Cyprus became independent for a while, but later came under the influence of the Mamluks.

3. Ottoman Era

Ottomans invaded Cyprus and brought their own system. Daily life started to change. You could see a mix in culture, food, architecture and how people lived. The Ottoman era had a big impact on Cyprus because they ruled it for a long time. This era ended during the time of Abdulhamid.

4. British Period

After the Ottomans, the British took control of the island. They built roads, schools and introduced modern systems. But this also made people want independence more than before. This period ended with protests and chaos as people fought for their future.

5. Modern Republic of Cyprus

Cyprus became independent from British rule. But things didn't settle. Even Greek groups started fighting among themselves. In the end, the island got divided. That separation is still going on today. Both sides have moved forward in their own ways and built their own systems

6. Future of Cyprus

The future is about imagination. Maybe the people of Cyprus will build flying cars, maybe they'll go to space, or maybe robots will guide tourists around the cities. No one knows what will happen. What's possible is endless.

6. Development Tools and Workflow



- Programming Language: C#
- Design Tools: Unity UI system
- Assets: Mix of custom and Unity Asset Store items

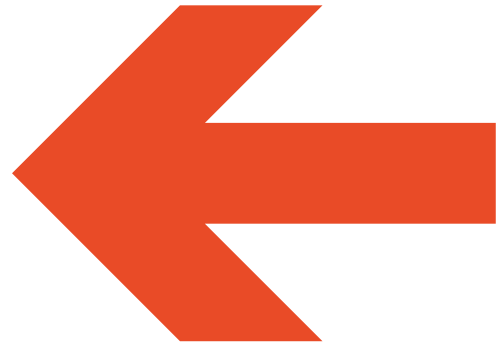
These are the tools I used for developing the game. Unity is a game engine that is free for students in many countries around the world. It is one of the most basic and important game development programs that you must learn if you want to create games with passion. Unity's design tools are built into the engine, and with its own UI system, you can design anything you imagine in a short period of time.

Unity also provides assets from its website. Whenever you want to try something or are curious if an idea would look good in your project, you can easily find free assets, test them, and see how they fit your game. If

you like them, you can even buy the full versions from the Unity Asset Store and improve your project even more.

The best part is that you don't need to be an expert to use Unity. It is beginner-friendly and there are lots of tutorials online to help you learn step by step. With Unity, you can focus on both visuals and gameplay, and it gives you the freedom to make your dream game real.

7.Challenges and Solutions



- UI scaling issues on multiple devices:
- Solved with Canvas Scaler. Unity provides auto scaling for canvas. I used it and fixed my problems in another devices. Before that, everything was looking weird on different screens. On some phones buttons were too big, on some they were out of the screen. After I used Canvas Scaler and set it to scale with screen size, all UI elements started looking normal. It really saved my time and now the game looks fine on small and big screens without any extra work.
- Score increment not working:
- Fixed using collectable script that works like coin. So I put coin in the start of the game, and you are collecting it every single time you start game so game keeps track of your runs with coin script. Before this, I couldn't really track how many times the player played the game. I wanted something simple and fun, so using coin logic

- was easy and made sense. I also added a counter system that updates with the coin, so it gives motivation to keep running and getting a better high score.
- Button navigation to history panel:
- Fixed with SceneManager and put variable for every scene. At the start, buttons weren't working or going to the wrong places. Then I made a small script for buttons to load the scenes or enable the panels. History panel opens with one click now, and other panels too. It helped the menu feel smoother and more like a finished game.
- Character getting out of map:
- I put limitation to character's limit of going X, Y, Z so character is not going out of bounce. Sometimes player would fall off the map or just run into
-
- empty space. That was looking bad. So I made a system to keep the player inside a certain area. I also made invisible walls to make sure even if you try, you can't leave the main map. This helped the gameplay feel more controlled and clean.
- Animation Problem:
- I had problem with character not doing its animation. I took animations from Mixamo and made a script that uses animator in Unity without player seeing it. Script GetComponent<Animator> was savior in this action. Without this, my character looked frozen or buggy. After connecting animations to the states (like running, jumping), it started looking alive. Now when the player runs, jumps or dies, you can see the animation playing smoothly.
-
-

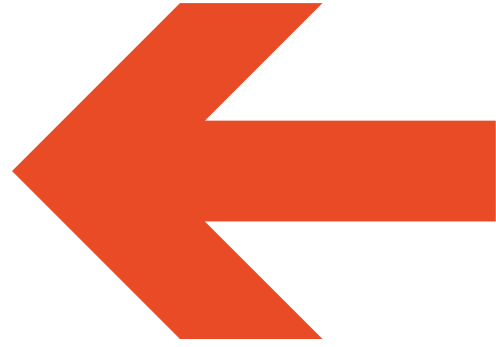
- Game being easy:
- When segments auto generated I had to be careful of game not forcing you to die. So I made one huge map that is hard to finish, so gameplay time and difficulty of the game increased by it. Before, the game was finishing too early or was boring. Now with the longer map and more obstacles, the game feels more like a challenge. I also made sure it's not too hard from the beginning, but it gets harder while you play more. That keeps players interested and trying to go further each time.

8. Testing And Feedback



I tested the game with my friends and game feedback was: my friends appreciated the educational aspect and suggested smoother transitions and sound effects for history section of the game, while first tests game was too easy so we had to change segment generation.

9.Future Improvements

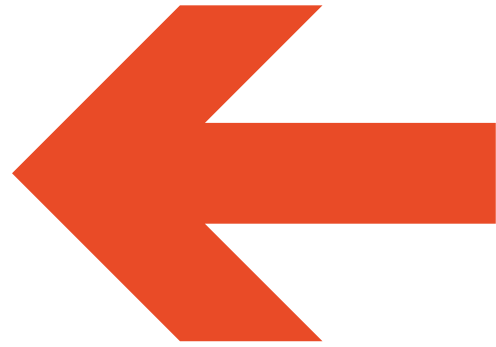


- Unlockables Based on History Milestones
-
- We got a second character, Kamil. If you read every part of the history section and pass the secret quiz, you might unlock a hidden character to play instead of Necmi. Just a fun little bonus for the lore nerds.
- Randomized Map / Endless Mode
-
- Once you finish the main game, we might unlock an Endless Mode where you can play on randomized terrain. Not in yet, but 100% something we can add later — replayability boost.
- Audio Narration for History
- We don't have rich sound design yet, especially in the history parts. Maybe in the future we throw in some narration to make it more alive. Depends on time/resources, but it's on the radar.
- Leaderboard for Run Counts

- Would be cool to show a leaderboard based on how many runs it took you to finish the game — or maybe how many times you've beaten it total. Gives speedrunners and perfectionists something to chase.
- Interactive History w/ AI
-

Later down the line, we could make the history section interactive — like, you import requests or ask Gemini AI stuff about Cyprus history, and it throws questions back at you. Would make learning way more dynamic.

10. Conclusion



CyprusRun has a unique gameplay style that no other game really has. It's a solo-developed project that tries to mix fun and learning in a different way. In future versions, the plan is to bring more ways to learn through AI, add leaderboards so people can compete, and include a lot more stuff that goes beyond what you'd expect.

11.Screenshots






```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;

@ Unity Script (2 asset references) | 1 reference
public class PlayerMovement : MonoBehaviour
{
    [SerializeField] GameObject playerAnim;

    public float playerSpeed = 6;
    public float horizontalSpeed = 3;
    public float rightLimit = 5.5f;
    public float leftLimit = -5.5f;
    public float jumpSpeed = 5f;
    public float jumpHeight = 3f;
    private float initialYPosition;
    private bool isMovingUp = false;
    private bool isFalling = false;
    private float targetYPosition;

    @ Unity Message | 0 references
    void Start()
    {
        initialYPosition = transform.position.y;
    }

    @ Unity Message | 0 references
    void Update()
    {
        // Forward movement
        transform.Translate(Vector3.forward * Time.deltaTime * playerSpeed, Space.World);

        // Left movement
        if (Input.GetKey(KeyCode.A) || Input.GetKey(KeyCode.LeftArrow))
        {
            if (this.gameObject.transform.position.x > leftLimit)
            {
                transform.Translate(Vector3.left * Time.deltaTime * horizontalSpeed);
            }
        }

        // Right movement
        if (Input.GetKey(KeyCode.D) || Input.GetKey(KeyCode.RightArrow))
        {
            if (this.gameObject.transform.position.x < rightLimit)
            {
                transform.Translate(Vector3.right * Time.deltaTime * horizontalSpeed);
            }
        }

        if (Input.GetKeyDown(KeyCode.Space) && !isMovingUp && !isFalling && Mathf.Approximately(transform.position.y, initialYPosition))
        {
            isMovingUp = true;
            targetYPosition = initialYPosition + jumpHeight;

            // Play jump animation
            playerAnim.GetComponent<Animator>().Play("Jump");
        }

        // Handle Moving Up
        if (isMovingUp)
        {
            transform.Translate(Vector3.up * jumpSpeed * Time.deltaTime);
            if (transform.position.y >= targetYPosition)
            {
                transform.position = new Vector3(transform.position.x, targetYPosition, transform.position.z);
                isMovingUp = false;
                isFalling = true; // Immediately start falling after reaching the apex
            }
        }

        // Handle Falling
        else if (isFalling)
        {
            transform.Translate(Vector3.down * jumpSpeed * Time.deltaTime);
            if (transform.position.y <= initialYPosition)
            {
                transform.position = new Vector3(transform.position.x, initialYPosition, transform.position.z);
                isFalling = false; // Reset state, ready for a new jump
            }
        }
    }
}

```

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.SceneManagement;

@ Unity Script (168 asset references) | 0 references
public class CollisionDetect : MonoBehaviour
{
    [SerializeField] GameObject thePlayer;
    [SerializeField] GameObject playerAnim;
    [SerializeField] AudioSource collisionFX;
    [SerializeField] GameObject mainCam;
    [SerializeField] GameObject fadeOut;

    private bool hasCollided = false;

    @ Unity Message | 0 references
    void Update()
    {
        if (Input.GetKeyDown(KeyCode.Space) && !hasCollided)
        {
            StartCoroutine(JumpAnimation());
        }
    }

    1 reference
    IEnumerator JumpAnimation()
    {
        playerAnim.GetComponent<Animator>().Play("Jump");
        yield return new WaitForSeconds(1f);
        playerAnim.GetComponent<Animator>().Play("Running");
    }

    @ Unity Message | 0 references
    void OnTriggerEnter(Collider other)
    {
        hasCollided = true;
        StartCoroutine(CollisionEnd());
    }

    1 reference
    IEnumerator CollisionEnd()
    {
        collisionFX.Play();
        thePlayer.GetComponent<PlayerMovement>().enabled = false;
        playerAnim.GetComponent<Animator>().Play("Stumble Backwards");
        mainCam.GetComponent<Animator>().Play("CollisionCam");
        yield return new WaitForSeconds(2);
        fadeOut.SetActive(true);
        yield return new WaitForSeconds(2);
        SceneManager.LoadScene(0);
    }
}

```

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;

@ Unity Script (2 asset references) | 1 reference
public class MasterInfo : MonoBehaviour
{
    public static int coinCount = 0;
    [SerializeField] GameObject coinDisplay;

    @ Unity Message | 0 references
    void Update()
    {
        coinDisplay.GetComponent<TMPPro.TMP_Text>().text = "RUNS: " + coinCount;
    }
}

```


12. References



Your paragraph text
<https://assetstore.unity.com/packages/3d/environments/landscapes/lowpoly-environment-pack-99479>

<https://assetstore.unity.com/packages/3d/environments/old-sea-port-environment-36897>

<https://assetstore.unity.com/packages/3d/environments/sci-fi/sci-fi-futuristic-environment-pack-v2-0-246983>

<https://assetstore.unity.com/packages/3d/environments/urban/simple-city-pack-plain-100348>

[https://en.wikipedia.org/wiki/Cyprus Hill, Sir George. A History of Cyprus, Vol. 4.](https://en.wikipedia.org/wiki/Cyprus_Hill,_Sir_George.A_History_of_Cyprus,_Vol.4)

[https://en.wikipedia.org/wiki/History of Cyprus#Ottoman Cyprus \(1571%E2%80%931878\)](https://en.wikipedia.org/wiki/History_of_Cyprus#Ottoman_Cyprus_(1571%E2%80%931878))

[https://en.wikipedia.org/wiki/Millet \(Ottoman Empire\).](https://en.wikipedia.org/wiki/Millet_(Ottoman_Empire))

[https://en.wikipedia.org/wiki/Ottoman conquest of Cyprus](https://en.wikipedia.org/wiki/Ottoman_conquest_of_Cyprus) <https://books.openedition.org/efa/3081>

<https://www.mixamo.com/#/?page=2&type=Character>

<https://assetstore.unity.com/packages/p/environmental-asset-pack-170036>

<https://assetstore.unity.com/packages/3d/environments/hand-painted-nature-kit-lite-69220>

<https://assetstore.unity.com/packages/3d/environments/low-poly-environment-315184>

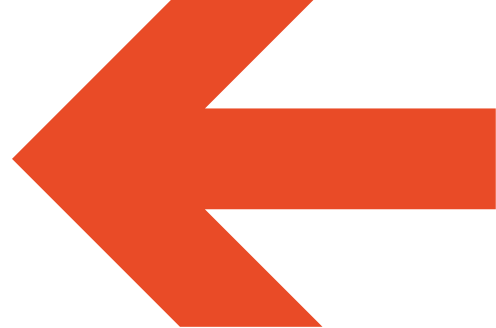
<https://assetstore.unity.com/packages/3d/environments/lowpoly-environment-nature-free-medieval-fantasy-series-187052>

<https://cypustravellerguide.com/the-city-kingdoms-of-ancient-cyprus/>

https://en.wikipedia.org/wiki/Ten_city-kingdoms_of_Cyprus

<https://www.metmuseum.org/essays/ancient-greek-colonization-and-trade-and-their-influence-on-greek-art>

2.1 Page Setup



Type the manuscript on one side of the standard-sized heavy white bond paper, (A4 format 297x210, portrait orientation). The page setup margins are: (2.5 cm) at the top, bottom and right, whereas the left margin is (3.5 cm). The reason for increasing the left margin is that the project will be bound and enough space should be left so as not to cover the text.

In addition to setting up the margins, the following line spacing settings should be used uniformly throughout the project:

One and half (1.5) spacing throughout the project

One and half (1.5) spacing after every line of the title, headings, quotations, references, figures, tables, ...etc

Page numbering should be consecutive. Type the numbers in the bottom center using Arabic numerals for the main parts of the project (i.e. Introduction, the chapters and Conclusion).

Arrange the pages of the manuscript as follows:

Title page should not be numbered

Acknowledgment should be numbered with (i)

Abstract page should be numbered with (ii)

Contents page(s) should be numbered starting with (iii)

Text starts on a new page numbered (1)

Acknowledgment should be numbered with (i)

Abstract page should be numbered with (ii)

Contents page(s) should be numbered starting with (iii)

Text starts on a new page numbered (1)

Punctuation and Spacing

Punctuation marks provide clearer meaning to the sentences and paragraphs. When using punctuation marks, make sure your use of spacing is applied as follows:

- One space to appear after all punctuation marks
- Exceptions:
 - No space after internal periods in abbreviations (a.m., i.e., U.S.)
 - No space after the colon in ratios (6:1)
 - No space before or after hyphens (trail-by-trail analysis)

B5. Abbreviations

An abbreviation is a short form of a word that can be obtained by omitting some letters of the end of the word. There are also contractions, which are also short forms of words, but obtained by omitting some middle letters of the word.

- Examples of frequently used common abbreviations are.
- ca. – (circa) meaning "at or near a given date"
- cf. – (confer, compare, or consult)
- i.e. – (id est, "that is")
- et al. – (et alii) meaning "and others", to be used for multiple authors, etc.
- e.g. – (exempli gratia) meaning "for example"
- n. – ("note")
- Passim – ("here and there")
- Sic – ("thus") italicized in brackets [] to indicate error in original quote
- n.d. – ("no date") if no date is given for publication, letter, etc.

2.2 Overview



All software engineering students are required to give oral presentations at the end of their internship project. The audience usually consists of the internship jury, which may include academic staff and fellow students.

The presenting student is allowed to schedule their presentation only after their internship supervisor has confirmed that the internship period has been successfully completed and the project outcomes are ready to be presented.

The student is expected to summarize the internship topic, the responsibilities undertaken, and the conclusions or deliverables achieved. The student may also provide insight into how the internship experience developed over time, such as:

- What was the main objective of the project?
- What challenges occurred?
- What unexpected issues occurred during the project?
- From the student's point of view, what are the strengths of doing the internship project?
- What aspects of the project could be improved further?

The internship jury are invited here to ask questions about the project scope, the student's technical contribution to the problem, problem-solving methods, and the outcomes produced.

There is no time limit for the presentation, but it is still recommended to keep it clear, structured, and well-delivered.

2.2Bibliography



- A guide to academic writing for social sciences. Prepared by the Department of Sociology, University of Minnesota. Retrieved May 8, 2002 from the World Wide Web: "A guide for writing research papers based on Modern Language Association's documentation. Prepared by the Humanities Department and Arthur C. Banks, Jr., Library Capital Community College. Hartford, Connecticut. Retrieved May 31, 2000 from the World Wide Web: "Your paragraph text".
- Guide to grammar and writing. Retrieved November 24, 2000 from the World Wide Web: "Your paragraph text".
- A research guide for students. Retrieved July, 1998 from the World Wide Web: "Your paragraph text".
- The Basic Elements of English. An interactive guide to grammar. Retrieved July 15, 1999 from the World Wide Web: "Your paragraph text".
- A brief guide to writing philosophy papers. Retrieved November 26, 1999 from the World Wide Web: "Your paragraph text".
- A guide for writing research papers, APA style. Retrieved December 6, 1999 from the World Wide Web: "Your paragraph text".
- Academic writing. Retrieved February 10, 1999 from the World Wide Web: "Your paragraph text".
- Khashman A., Fakhraddin M., Bektaş Ş., "Graduate Studies: A Complete Reference", NEU Press, 2001.
- Khashman A., "Automatic Detection of Military Targets Utilizing Neural Networks and Scale Space Analysis", NATO Information Systems Technology Panel (IST) Symposium, October 2000, Istanbul, Turkey.

- Mamedov F. and Özerdem Ö., "Handbook BE400", NEU Press, 2001.
- Pryor J. Guidelines on writing philosophy papers. Retrieved April 27, 1999 from the World Wide Web: "Your paragraph text".
- .
- Purdue University Online Writing Lab (OWL). Guide to APA Formatting and Style. Retrieved November 15, 2003 from the World Wide Web: "A guide for writing research papers based on Modern Language Association's documentation. Prepared by the Humanities Department and Arthur C. Banks, Jr., Library Capital Community College. Hartford, Connecticut. Retrieved May 31, 2000 from the World Wide Web: "Your paragraph text".
- Guide to grammar and writing. Retrieved November 24, 2000 from the World Wide Web: "Your paragraph text".
- A research guide for students. Retrieved July, 1998 from the World Wide Web: "Your paragraph text".
- The Basic Elements of English. An interactive guide to grammar. Retrieved July 15, 1999 from the World Wide Web: "Your paragraph text".
- A brief guide to writing philosophy papers. Retrieved November 26, 1999 from the World Wide Web: "Your paragraph text".
- A guide for writing research papers, APA style. Retrieved December 6, 1999 from the World Wide Web: "Your paragraph text".