Project Report On

DesignEZ

Submitted in fulfillment of the requirements for the award of the degree of

Bachelor of Technology

in

Computer Science and Engineering

By

Pranav Sridhar Natarajan (U2003161)
Rony Mons (U2003175)
S Gokul Raj (U2003179)

Under the guidance of

Dr. Mary Priya Sebastian

Department of Computer Science and Engineering
Rajagiri School of Engineering & Technology (Autonomous)
(Parent University: APJ Abdul Kalam Technological University)
Rajagiri Valley, Kakkanad, Kochi, 682039

April 2024

CERTIFICATE

This is to certify that the project report entitled ***"DesignEZ"*** *is a bonafide record of the work done by* ***Pranav Sridhar Natarajan (U2003161)*** *,* ***Rony Mons (U2003175)*** *and* ***S Gokul Raj (U2003179)*** *submitted to the Rajagiri School of Engineering & Technology (RSET) (Autonomous) in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology (B. Tech.) in Computer Science and Engineering during the academic year 2023-2024.*

Dr. Mary Priya Sebastian
Project Guide
Associate. Professor
Dept. of CSE
RSET

Ms. Anitha John
Project Co ordinate
Asst. Professor
Dept. of CSE
RSET

Dr. Preetha K. G.
HoD
Dept. of CSE
RSET

ACKNOWLEDGMENT

I wish to express my sincere gratitude towards **Prof. Dr. P.S. Sreejith**, Principal of RSET, and **Dr. Preetha K.G**, Head of the Department of Computer Science and Engineering for providing me with the opportunity to undertake my project, ”DesignEZ”.

I am highly indebted to my project coordinators, **Ms. Anitha John**, Asst. Professor, Department of CSE, for her valuable support.

It is indeed my pleasure and a moment of satisfaction for me to express my sincere gratitude to my project guide **Dr. Mary Priya Sebastian**, Associate Professor, Department of CSE, for her patience and all the priceless advice and wisdom she has shared with me.

Last but not the least, I would like to express my sincere gratitude towards all other teachers and friends for their continuous support and constructive ideas.

Pranav Sridhar Natarajan

Rony Mons

S Gokul Raj

Abstract

The project creates a system for automating residential building model creation with the Unity 3D game engine using procedural generation. Users provide parameters like plot dimensions, and the system's algorithm constructs a complete 3D model with basic room layouts. This approach offers significant advantages in terms of development efficiency and design exploration. Procedural generation allows for the rapid creation of numerous unique models, fostering design inspiration and accelerating content creation workflow in game development, architectural visualization, and urban planning simulations. While the current implementation relies on room sizes as percentages of the total plot area, limiting the achievable variety of architectural styles, this research demonstrates the potential of procedural generation for efficient residential building model creation. Future work will focus on expanding the algorithm's capabilities to generate a wider range of architectural styles.

In conclusion, "DesignEZ" is an innovative initiative that harmonizes user input, procedural generation, and Unity 3D capabilities. This fusion results in a sophisticated tool for residential building design, marking a paradigm shift in architectural creativity and effectiveness — from plot analysis to 3D modeling and interior layout creation.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The introduction chapter serves as the gateway to the "DesignEZ" project, offering readers a comprehensive overview of its objectives, methodologies, and significance in the realm of architectural design. It lays the foundation for a detailed exploration of the innovative approach taken in this transformative initiative.

1.1 Background

The contextual backdrop against which "DesignEZ" unfolds is marked by a discernible shift in conventional architectural design paradigms. Historically, designers have grappled with the intricate balance between imaginative vision and the pragmatic constraints inherent in construction. Against this historical narrative, the advent of technology, particularly within the Unity 3D framework, presents a distinct opportunity to revolutionize the architectural design process. The evolution of architectural tools over time lays the groundwork for "DesignEZ," as it endeavors to harness procedural generation methodologies to usher in efficiency, creativity, and enhanced user engagement in the forefront of architectural innovation. In an era where the fusion of creativity and technology holds paramount significance, "DesignEZ" emerges as a pivotal force in redefining the conceptualization and creation of residential spaces.

1.2 Problem Definition

The problem that existed which brought the need to come up with this was there are tools that only give the 2D drawings of buildings,building generation is a time consuming process,it requires intense communication skills.

1.3 Scope and Motivation

The scope of "DesignEZ" encompasses a comprehensive procedural generation approach to residential building design. Users actively contribute plot area data, triggering a meticulous analysis to ascertain the most viable construction space. This phase not only emphasizes optimal spatial utilization but also lays the foundation for the subsequent generation of sophisticated and purposeful floor plans. The procedural generation algorithms employed in "DesignEZ" ensure the creation of diverse and intricate floor plans within the identified feasible areas. The resulting architectural blueprints represent a harmonious synthesis of imaginative design and meticulous functional considerations, promising to redefine the scope of architectural creativity in the design of residential spaces.

The motivation behind "DesignEZ" lies in inspiring active user engagement and creativity during the planning and construction phase of residential buildings. By encouraging individuals to contribute plot area data, the project aims to foster a sense of ownership and imaginative input in the design process. The procedural generation methodologies not only streamline the design workflow but also provide users with a tool that seamlessly translates their creative ideas into tangible and realistic 3D models. Ultimately, the project is motivated by a vision to empower users, both professionals and enthusiasts, in shaping their ideal residential spaces, marking a significant paradigm shift in the approach to architectural design.

1.4 Objectives

- Develop methods for optimizing spatial utilization in residential design, emphasizing effective layouts based on user-provided plot area data.
- Utilize procedural generation techniques within Unity 3D to create diverse and intricate floor plans within identified feasible areas.
- Seamlessly blend user input with generation techniques, fostering collaborative and personalized architectural design on a user-friendly platform.
- Leverage Unity 3D capabilities to construct realistic 3D models of residential spaces, providing users with immersive visualizations of their envisioned designs.

- Extend the tool's capabilities to generate basic interior layouts, catering to both professionals and enthusiasts engaged in the design process.
- Create a user-friendly platform for enthusiasts to explore and refine their creative ideas, making architectural design accessible to a wider audience.
- Address algorithmic complexity challenges, ensuring the efficiency and reliability of procedural generation throughout the architectural design process.

1.5 Challenges

Despite its innovative approach, "DesignEZ" confronts challenges such as algorithmic complexity, devising effective user engagement strategies, and ensuring the seamless integration of procedural generation methodologies within the Unity 3D framework.

1.6 Assumptions

Operating under the assumption that active user participation enhances design diversity, and procedural generation algorithms can effectively balance creativity and functionality, "DesignEZ" sets the stage for an exploration into the assumptions guiding the project's development.

1.7 Societal / Industrial Relevance

- Empowers individuals in actively contributing to their residential design, fostering a sense of ownership and creativity.
- Facilitates collaborative design practices, encouraging interaction and feedback from diverse user groups.
- Enhances accessibility to architectural design, making it user-friendly for both professionals and enthusiasts.
- Integrates technological advancements into architectural creativity, aligning with the contemporary trend of digital innovation.

- Streamlines the architectural design process, increasing efficiency and effectiveness for professionals in the industry.
- Broadens the scope of user engagement in architectural projects, contributing to a more inclusive and diverse design landscape.
- Offers a valuable tool for educational purposes, allowing students to explore architectural concepts in a practical and engaging manner.
- Aligns with the current emphasis on sustainable and efficient spatial utilization, addressing modern societal and environmental concerns.
- Paves the way for advancements in procedural generation methodologies within architectural design, contributing to industry innovation.
- Provides a platform for interdisciplinary collaboration, bringing together professionals, enthusiasts, and technology in the architectural realm.

1.8 Organization of the Report

This report is meticulously organized to offer readers a structured journey through the "DesignEZ" project. Following this introduction, subsequent chapters delve into the methodology, implementation, results, and conclusions, providing a comprehensive understanding of the project's development.

In conclusion, this introduction chapter serves as a compass for navigating the intricate landscape of "DesignEZ." It introduces the motivations, objectives, and challenges that will be explored in detail in the upcoming chapters, setting the stage for a comprehensive examination of this pioneering initiative in architectural design.

Chapter 2

Literature Survey

2.1 A Declarative Tile-Based Approach for Procedural Generation of Architecture[1]

2.1.1 Introduction

Procedural Content Generation (PCG) has emerged as a critical tool in digital content creation, particularly in architecture. The paper titled *"A Declarative Tile-Based Approach for Procedural Generation of Architecture"* presents an innovative method that combines declarative and comprehensive characteristics. This review aims to highlight the key advantages of this approach outlined in the document *"gokul_Main (2).pdf"*, emphasizing their significance in architectural generation.

2.1.2 Advantages of the Presented Approach

sectionDeclarative and Comprehensive Nature The method's reliance on architectural profiles rather than grammars or training sets enhances adaptability within a generic tile-solving framework, fostering creative freedom.

Configurability and Diversity

The approach's ease of configuration allows for the generation of diverse architectural structures, catering to various design requirements and preferences.

Tunability for Variability

The method's tunability enables the generation of architectural structures across significant ranges of density and repetitiveness, providing control over creative diversity.

Potential for Urban Environments in Game Contexts

The method's adaptability and configurability make it suitable for generating urban environments in game contexts, aligning well with the needs of indie game development, such as those seen in Minecraft.

2.2 A New Reconstruction Method for 3D Buildings from 2D Vector Floor Plan[2]

2.2.1 Introduction

In the paper titled *"A New Reconstruction Method for 3D Buildings from 2D Vector Floor Plan,"* Junfang Zhu, Hui Zhang, and Yamei Wen address the challenge of reconstructing three-dimensional (3D) models from two-dimensional (2D) architectural floor plans. Published in the Computer-Aided Design and Applications journal in 2014, the paper introduces a method to analyze 2D floor plans, extract topological and semantic information, and reconstruct precise 3D models.

2.2.2 Key Contributions

The authors propose a novel approach that utilizes a shape-opening graph and loop searching method for efficient reconstruction. This method is capable of handling complex floor plans containing thousands of geometric primitives within a short timeframe.

2.2.3 Advantages of the Paper

The paper's contributions extend beyond its innovative methodology. It offers a rapid reconstruction process, incorporating both geological and semantic information into the 3D models. Additionally, the creation of a topological model facilitates the analysis of space connectivity and relationships, providing valuable insights for architecture, urban planning, and virtual reality applications.

2.3 Effective Interaction with Virtual Building Environments[3]

2.3.1 Introduction

The paper titled *"Effective Interaction with Virtual Building Environments"* by Rick Lewis and Carlo H. Se´quin represents a seminal contribution to the field of computer graphics, particularly in the realm of creating realistic virtual building environments. Published as a Master's Thesis in the Technical Report UCB/CSD-95-886 by the Computer Science Division (EECS) at the University of California, Berkeley in 1995, the paper addresses the challenges inherent in generating detailed three-dimensional (3D) building models from two-dimensional (2D) representations and proposes innovative solutions to enhance user interaction within these virtual environments.

2.3.2 Key Contributions

At the heart of the paper lies a novel method for constructing detailed frame models tailored to various portal types. This methodology enables the incorporation of components that cannot be accurately represented using traditional extrusion techniques, thereby enhancing the realism and fidelity of virtual building environments. A noteworthy aspect of this approach is its ability to automatically position and size inserted door models based on an analysis of the input floor plan, streamlining the modeling process and ensuring geometric accuracy.

2.3.3 Limitations and Proposed Solutions

The authors meticulously analyze the limitations associated with conventional extruded prismatic shapes in accurately capturing the structural complexity of buildings. Through detailed examples, such as the arches adorning the seventh floor of Soda Hall, they underscore the inadequacies of simplistic modeling approaches. To address these limitations, the paper advocates for the inclusion of additional detailed components, such as arches and intricate architectural features, to improve the realism and fidelity of virtual building representations.

2.3.4 Related Work and References

In addition to presenting their innovative methodology, Lewis and Se´quin contextualize their work within the broader research landscape by referencing related studies in computer graphics and virtual building environments. By citing works on image realism, interactive update rates, and user interface design tailored for architects and engineers, the authors provide a comprehensive overview of the state-of-the-art techniques and methodologies in the field, enriching the discussion surrounding their proposed method.

2.4 Automatic reconstruction of 3D building models from scanned 2D floor plans[4]

2.4.1 Introduction

This paper critically evaluates the methodologies employed in generating three-dimensional (3D) digital models of existing buildings. It concludes that no universally optimal approach exists for this task, emphasizing the necessity of considering end-user objectives and project constraints when selecting a method. Furthermore, the review enumerates various techniques, including photogrammetry, laser scanning, tagging, and leveraging preexisting information such as sketches and tape measurers, all aimed at creating 3D models of extant buildings. However, despite the breadth of available techniques, none seem to offer a cost-effective and reliable solution for this purpose.

Moreover, the article delineates the challenges associated with producing 3D models at reasonable costs, a significant impediment to the widespread adoption of Building Information Modeling (BIM) in renovation projects. It introduces a research endeavor aimed at developing methodologies for generating 3D building models from two-dimensional (2D) plans. The resultant prototype is capable of extracting information from 2D plans and generating Industry Foundation Classes (IFC)-compliant 3D models, encompassing fundamental components such as walls, openings, and spaces. Additionally, the paper underscores the advantages of employing BIM Information and Communication Technology (ICT) tools in building design practices, emphasizing their potential to enhance the efficacy of renovation designs.

Despite the myriad techniques explored, a definitive solution remains elusive in the quest for efficient and reliable generation of 3D digital models of existing buildings. The challenges of cost-effectiveness and accuracy persist, hindering widespread adoption in renovation projects. Nonetheless, ongoing research endeavors offer promising avenues for overcoming these hurdles. By developing methodologies that bridge the gap between 2D plans and 3D models, there is a potential to revolutionize building design practices. Embracing these advancements in Building Information Modeling (BIM) ICT tools holds the key to unlocking transformative possibilities in the field of architectural visualization.

2.5 Automatic Generation of 3D Building Models from Architectural Drawings Using 3DPlanNet Ensemble[5]

2.5.1 Introduction

This paper presents an innovative approach for automatically generating three-dimensional (3D) building models from two-dimensional (2D) architectural drawings. Employing a combination of data-driven and rule-based methodologies, the authors propose a technique that achieves significant accuracy in wall restoration and precise size determination, surpassing 97% in creating 2D drawings devoid of dimension information.

2.5.2 Key Contributions

Comparative analysis with prior studies demonstrates the efficacy of the proposed approach in achieving comparable accuracy levels despite utilizing a smaller training dataset. This success is attributed to meticulous annotation practices in the training data and strategic sequencing of wall detection. Moreover, the paper elucidates the method's enhanced accuracy in identifying wall junctions, as validated through empirical experiments.

2.5.3 Implications and Future Research

The paper concludes by delineating the diverse applications of the proposed method, spanning urban planning, architecture, and interior design. Furthermore, the authors advocate for future research endeavors aimed at refining the method's performance and broadening its applicability across various image types. This forward-looking perspective underscores the paper's contribution to the ongoing advancement of automated 3D building modeling techniques.

2.5.4 Advantages

It introduces an innovative methodology for automatically generating 3D building models from 2D architectural drawings, showcasing impressive accuracy in wall restoration and precise size determination. Moreover, the proposed method exhibits potential utility across a spectrum of applications, emphasizing its relevance and significance in the field of computer-aided design and architectural visualization.

2.6 Floor Plan Generation as an Optimization problem[6]

2.6.1 Introduction

This paper presents a method for generating diverse floor plans tailored for residential projects during their initial developmental stages. One of the principal strengths of this method lies in its simplicity and compatibility with most architectural software, rendering it highly customizable and adaptable to specific project needs.

Furthermore, the method leverages simulated annealing as an optimization algorithm to explore the problem space comprehensively, facilitating the attainment of optimal solutions. This approach allows for a thorough exploration of the problem space, resulting in diverse and compelling floor plan designs.

2.6.2 Prior Research

In addition to outlining the proposed method, the paper provides a comprehensive overview of prior research on layout generation techniques, including the utilization of optimization strategies for generating layouts of multiple apartments within a rectangular floor space. This contextualization enriches the understanding of the method proposed in the paper and facilitates comparisons with previous approaches.

2.6.3 Advantages

- **Simplicity and Customizability**: The method is easily implementable in most architectural software, offering flexibility and adaptability to meet specific project requirements.
- **Utilization of Simulated Annealing**: By employing simulated annealing as an

optimization algorithm, the method ensures a thorough exploration of the problem space, leading to diverse and optimal floor plan designs.

2.7 A Markov Decision Process Workflow for Automating Interior Design[7]

2.7.1 Introduction

The paper offers a comprehensive exploration of various facets within the realm of interior design and decision-making processes, with a notable emphasis on leveraging technology and artificial intelligence (AI) systems to enhance efficiency and cost-effectiveness.

2.7.2 Key Contributions

One significant aspect discussed in the paper pertains to the utilization of letter-sized papers in indoor scenes as a mechanism for upscaling the entire scene through image processing and computer vision algorithms. Additionally, the paper introduces a workflow for expeditiously documenting as-built conditions using a video-to-point cloud (PCD) method, catering to clients with limited knowledge of modeling and documentation while concurrently reducing operational costs.

Moreover, the paper delves into the integration of AI systems in interior design decision-making, particularly through the application of Markov decision processes (MDP) to address architectural and design challenges. This innovative AI system exhibits the capability to comprehend its environment and emulate cognitive functions of interior designers, facilitating informed design decisions.

2.7.3 Advantages

- **Technological Innovation**: The paper showcases innovative approaches, such as utilizing letter-sized papers for scene upscaling and employing video-to-PCD methods for efficient as-built documentation, demonstrating the integration of technology to streamline interior design processes.

- **AI Integration**: By leveraging Markov decision processes (MDP), the paper highlights the potential of AI systems in solving complex architectural and interior design decision problems, thereby enhancing decision-making processes and design outcomes.

- **Cost-Effectiveness and Efficiency**: The proposed workflows and AI integration offer the promise of significant cost savings and improved efficiency in interior design practices, reflecting a commitment to optimizing resource utilization and enhancing overall project performance.

Chapter 3

Requirements

3.1 Hardware and Software Requirements

Minimum Requirements:

- Processor: Quad-core processor (Intel Core i5 or equivalent).
- Memory: 8 GB RAM.
- Storage: 256 GB SSD.

Software Requirements:

- Unity 3D: Version 2020.3 LTS or later.
- Unity Hub for version management.
- Unity Editor for project development.
- Programming Language: C# for Unity scripting.

Chapter 4

System Architecture

In this chapter, we will address the structure of the project and explore how different systems within it interact with each other, as well as the flow of data between these systems.

4.1 System Overview

Design EZ represents a Novel approach to residential building design, leveraging procedural generation techniques and user-friendly interfaces to streamline the entire process. This section provides a comprehensive overview of the project, including detailed architecture diagrams and a step-by-step outline of the entire process.

Process Outline

Design EZ follows a structured workflow, encompassing several stages from plot definition to layout generation and structural detailing. Below is a detailed outline of each stage:

- **Plot Dimension Input:**
  - Users provide length and width dimensions of the plot.
  - The system creates a corresponding grid and surface for interaction.
  - Interface prompts users for input and validates dimensions.
- **Plot Shape Drawing:**
  - Users draw the plot shape using mouse-based raycasting.
  - The system identifies closed shapes and calculates centroids.
  - Advanced algorithms analyze the shape to identify feasible areas.

  - Validation ensures accurate shape drawing and closed contours.

- **Feasible Area Identification:**
  - Starting from the centroid of the plot shape, the system initiates a recursive function to systematically explore and identify feasible cells within the polygon.
  - The recursive function checks each cell within the polygon, determining its feasibility based on predefined criteria.
  - If a cell is deemed feasible, the function recursively explores its adjacent cells to further identify viable areas while respecting the boundaries of the polygon.

- **Living Room Location Selection:**
  - Users select a cell within the feasible area for the living room.
  - Interface provides visual feedback and selection assistance.
  - Centroid calculation aids in initial placement for user convenience.

- **Optional Room Specification:**
  - Users specify areas for additional rooms like dining room, kitchen, and bedrooms.
  - Interface offers options for room selection and customization.
  - Constraints ensure user-defined areas align with feasibility considerations.

- **Layout Generation:**
  - The system divides the feasible area into public and private sections based on user preferences.
  - Percentage ratios dictate room sizes within the layout generation algorithm.
  - Public area rooms (living, dining, kitchen) generated using a modified flooding algorithm.
  - Private area rooms (bedrooms, bathrooms) generated based on predefined constraints.
  - Iterative process ensures optimal space allocation and layout coherence.

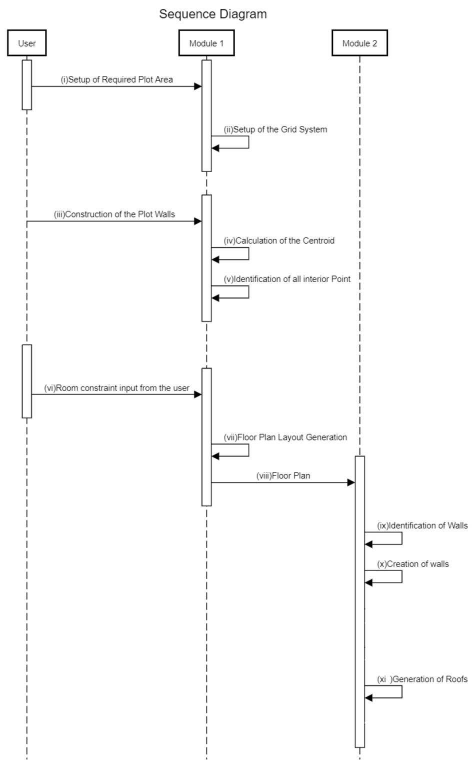


Figure 4.1: Sequence Diagram

- **Wall and Roof Generation:**
  - Walls are generated between rooms and along boundaries within the feasible area.
  - Roofs are placed above feasible cells, ensuring coverage and structural integrity.
  - Algorithmic logic prevents overlapping and ensures correct placement.
  - Rendering module visualizes generated walls and roofs for user feedback.

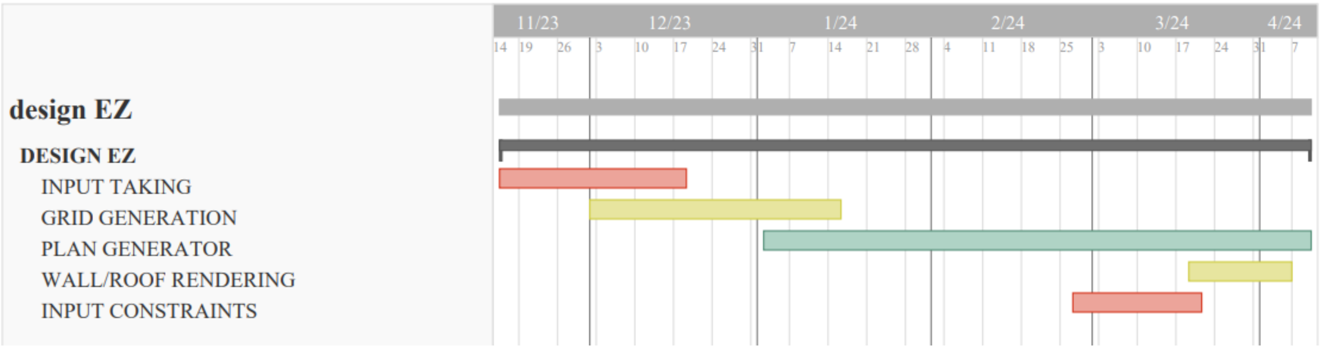
4.2 Module Division

- **User Interface Module:** This module handles user interactions, including input for plot dimensions, plot shape drawing, and room specification. It communicates with other modules to relay user input and receive feedback.
- **Algorithm Module:** The heart of the system, this module implements algorithms for plot shape recognition, area identification,Adaptive flood layout generation. It interacts with the database to process layout efficiently.

- **Database Module:** Responsible for managing data structures representing plots, feasible areas, room layouts, and structural elements. It ensures data integrity and accessibility across modules.

- **Rendering Module:** This module renders graphical representations of plots, layouts, walls, roofs, and other architectural elements for visualization and feedback. It communicates with the user interface module to provide real-time updates to users.

4.3 Work Schedule - Gantt Chart



4.4 Conclusion

Design EZ offers a comprehensive and efficient solution for residential building design, combining the power of procedural generation with user-centric interfaces. The detailed architecture and workflow outlined in this section provide a roadmap for the successful implementation and deployment of the system.

Chapter 5

System Implementation

This section delves into the technical aspects of implementing Design EZ, covering software development methodologies, programming languages, frameworks, and tools used. It also discusses the integration of various modules and algorithms, ensuring a seamless and efficient system implementation.

5.1 Constraint Design

This constraint set in Design EZ project offers predefined constraints for various room sizes within residential buildings. As numerical datasets for general guidelines are often unavailable, these constraints have been manually calculated with offsets from drawing of limited number of examples found on the internet. Utilizing percentage-based constraints makes the dataset easier to work with, providing clear guidelines for room sizes and enhancing the efficiency of the design process.

Room	Minimum (%)	Maximum (%)
Living Room	17	31
Dining Room	10	16
Single Bedroom	15	24
Kitchen	17	28
Attached Bathroom	5	7
Common Bathroom	6	8

Table 5.1: Dimensions of Rooms

5.2 Proposed Methodology/Algorithms

5.2.1 Plot Dimension Input

The project starts with the development of the surface to work with. Input is taken from the user to get the dimensions of the surface . Confirming the dimensions will generate the surface ready for the next stage.

5.2.2 UserWall

The algorithm 1 in the Design EZ project enables users to define wall placements interactively on a plain surface. Through mouse interactions, users can click and hold to initiate the creation of a wall by storing mouse coordinates as vertices in a Vector2 array. This process repeats until a closed polygon is formed upon mouse release.During each iteration, the algorithm calculates the midpoint between the start and end coordinates of the mouse and instantiates a userWall prefab at that position, effectively creating a wall based on the user's input. This intuitive approach empowers users to actively participate in the design process of inputting the accurate layout of the plot.

5.2.3 Feasible Area Identification

The Feasible Area Generation algorithm 2 constitutes a pivotal segment of the system architecture, focusing on identifying viable spaces within the user-defined plot boundaries. This module encompasses several interconnected components, primarily centered around the Feasible Area Identification process, which plays a crucial role in determining the feasible areas.

The architecture begins with the User Wall Module, responsible for processing user-drawn walls. This module extracts the vertices of the walls and computes the centroid of the resulting polygon. The centroid serves as the starting point for the feasibility identification process.

Employing algorithms like Flood Fill, this module systematically explores the polygonal boundaries, identifying feasible cells within. It recursively traverses adjacent cells, determining their feasibility based on predefined criteria. This process ensures that the generated design aligns with the identified feasible areas within the plot boundaries.

Algorithm 1 UserWallCreation(plainSurface)

procedure USERWALLCREATION(plainSurface)

Input: plainSurface: Plain surface where the user interacts to create walls.

Procedure:

Create an empty array of Vector2 to store vertices of the user-created polygon.

while Closed Polygon is not Formed **do**

1. On mouse button down event:
   (a) Perform a raycast from the mouse position to detect collision with the plain surface.
   (b) If the raycast hits the surface:
      i. Store the mouse coordinates as a Vector2 in the array.
2. On mouse button up event:
   (a) Perform a raycast from the mouse position to detect collision with the plain surface.

If the raycast hits the surface on mouse button up:

1. Calculate the distance between the stored start and end coordinates of the mouse.
2. Determine the midpoint between the start and end coordinates.
3. Instantiate the userWall prefab at the midpoint with its scale set to half of the calculated distance.

end while

end procedure

Algorithm 2 Feasible Area Identification

procedure FEASIBLEAREAIDENTIFICATION(User drawn walls)
  Store the vertices of the user-drawn walls in a 2D vector array
  Calculate the centroid of the polygon formed by the user-drawn walls
  Call Flood(cell)
end procedure
procedure FLOOD(cell)
  if cell is not a wall **then**
    Set the cell as feasible
    Call Flood(cell($x, y + 1$))
    Call Flood(cell($x, y - 1$))
    Call Flood(cell($x + 1, y$))
    Call Flood(cell($x - 1, y$))
  end if
end procedure

5.2.4 Layout Generation

The algorithm 3 systematically constructs the layout by iteratively placing and expanding rooms based on predefined constraints and user specifications. Beginning with the living room, the algorithm identifies neighbouring cells within the feasible area and randomly selects initial cells to initiate room placement. It then expands the room by adding adjacent cells, ensuring that the room size complies with specified percentage constraints relative to the total feasible area. This process is repeated for each room type, including dining rooms, bedrooms, kitchens, and bathrooms.

In addition to the primary layout generation process, the Design EZ project incorporates an additional constraint when optional room locations are selected by the user. This constraint influences the selection of neighbouring cells during the layout generation phase. The algorithm adheres to a predefined order for generating rooms, ensuring consistency and coherence in the layout hierarchy. To achieve this, the A* search algorithm is employed to find the closest neighbouring cell from the array to the next room cell in the specified order. This iterative process continues until the room cell becomes adjacent to the neighbouring cells in the array. By enforcing this constraint, the algorithm guarantees

Algorithm 3 LayoutGeneration(feasibleAreaMatrix)

procedure LayoutGeneration(feasibleAreaMatrix)
    Input: feasibleAreaMatrix: Matrix representation of the feasible area including room locations.
    Output: final layout matrix
    Initialize an empty layout matrix to represent the final layout.
    for each room type starting with the living room **do**
        Initialize an empty array to store neighbouring cells.
        while the number of cells in the room is less than the specified percentage **do**
            Select a random cell from the array with allocation value set false.
            Add the selected cell to the room.
            Add neighbouring cells of the selected cell to the neighbouring array.
        end while
        Generate a random number, either 1 or 2.
        if the random number is 1 **then**
            Select all neighbouring cells from the array that are in the same column as the selected cell.
        else if the random number is 2 **then**
            Select all neighbouring cells from the array that are in the same row as the selected cell.
        end if
    end for
    set allocation of all the selected state as true.
    Add all selected neighbouring cells to the room.
    Output the final layout matrix.
end procedure

proper connectivity between rooms, maintaining the integrity of the layout hierarchy and facilitating seamless transitions between different areas of the residential building.

5.2.5 Model Generation

The wall/roof generation algorithm 4 in Design EZ operates by iteratively scanning pairs of adjacent cells within the layout's feasible area. Upon detecting feasible cells, it instantiates roof prefabs at their coordinates. If the cells represent boundaries between different rooms, appropriate wall prefabs are instantiated to delineate the rooms. This systematic approach ensures the efficient generation of structural elements, including walls and roofs, while maintaining connectivity and coherence within the building layout.

5.3 User Interface Design

5.3.1 Layout Overview

The UI provides users with an overview of the current building layout, displaying rooms, walls, and other structural elements. Users can easily visualize the layout and make adjustments as needed.

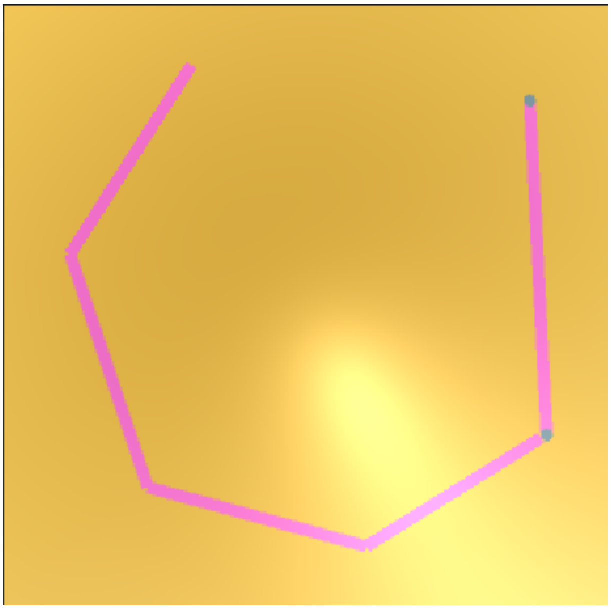


Figure 5.1: Project Interface for wall drawing

Algorithm 4 WallAndRoofGeneration(feasibleAreaMatrix)

procedure WALLANDROOFGENERATION(feasibleAreaMatrix)
    Input: feasibleAreaMatrix: Matrix representation of the land after layout generation.
    Define GameObjects:
    Define GameObjects for vertical walls, horizontal walls, and roof (1-meter square prefab).
    Repeat for Each Pair of Adjacent Cells in the Matrix:
    for each pair of adjacent cells (cell1, cell2) in the matrix **do**
        if cell1 and cell2 are adjacent and both are feasible cells **then**
            Instantiate the roof prefab at the position of cell1 with an offset of 0.
            if cell1 and cell2 represent a boundary between different rooms **then**
                if cell1 and cell2 are vertically adjacent **then**
                    Instantiate a horizontal wall prefab at the position of cell1 with an offset of 0.5.
                else if cell1 and cell2 are horizontally adjacent **then**
                    Instantiate a vertical wall prefab at the position of cell1 with an offset of 0.5.
                end if
            end if
        end if
    end for
    Output:
    Finalize the wall and roof generation process, providing structural elements for the building layout.
end procedure

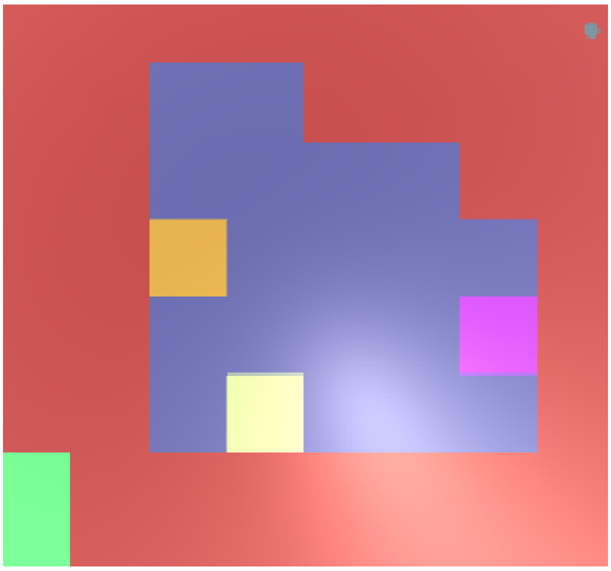


Figure 5.2: Project Interface for room location selection

5.3.2 User Controls

Intuitive controls are implemented to facilitate user interaction. For example, users can click and drag to place or move rooms, walls, and other elements. Keyboard shortcuts or gestures may also be incorporated to streamline common actions.

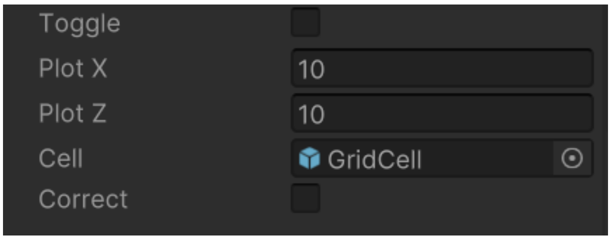


Figure 5.3: Input interface for entering the plot dimensions

5.3.3 Conclusion

the system implementation of Design EZ offers users a powerful and intuitive platform for creating customized residential building layouts. Through procedural generation techniques, user-defined constraints, and interactive tools, users can easily design layouts tailored to their preferences. The user interface enhances usability with clear layout overviews and intuitive controls. Overall, Design EZ provides a robust solution for residential building design, empowering users to unleash their creativity effectively.

Figure 5.4: Input interface for entering the room locations

Chapter 6

Results and Discussions

This section presents an evaluation of the outcomes and performance of the Design EZ project. This section aims to provide insights into the effectiveness of the implemented algorithms and features in generating residential building layouts and facilitating user interaction.

6.1 Result

The Project begins with a blank canvas, where users are prompted to define the dimensions of their plot. With a few clicks, generating a plot that mirrors their specifications.

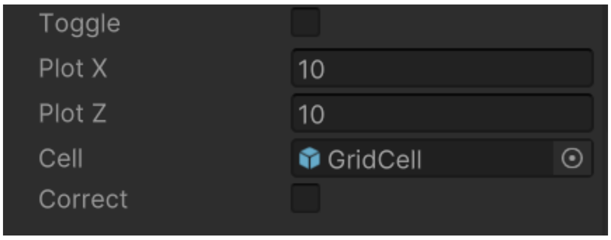


Figure 6.1: Stage 1

Next comes the creative phase. With the user input from the mouse, users define their building's layout, etching lines onto the canvas. These lines represents the boundary wall layout of the plot they need to work with.

As users complete their wall drawing, they're given the grid representation of the plan . A selection panel is provided, offering options to customize room locations. Amidst the choices, emphasis is placed on selecting the living room location, for setting the starting point of the algorithm.

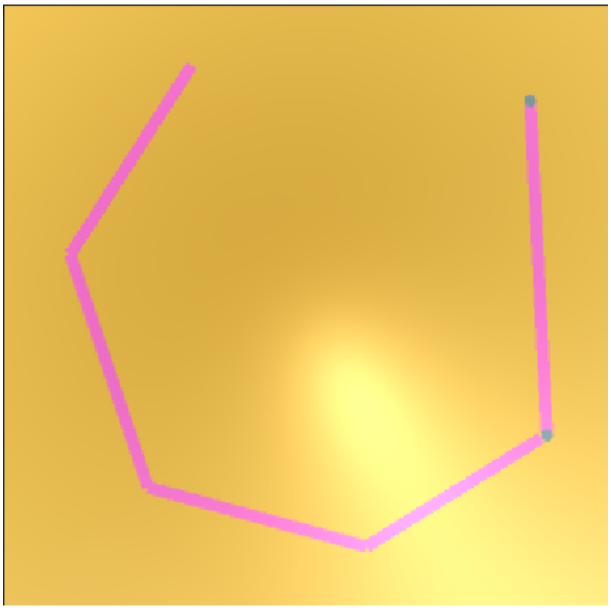


Figure 6.2: Stage 2

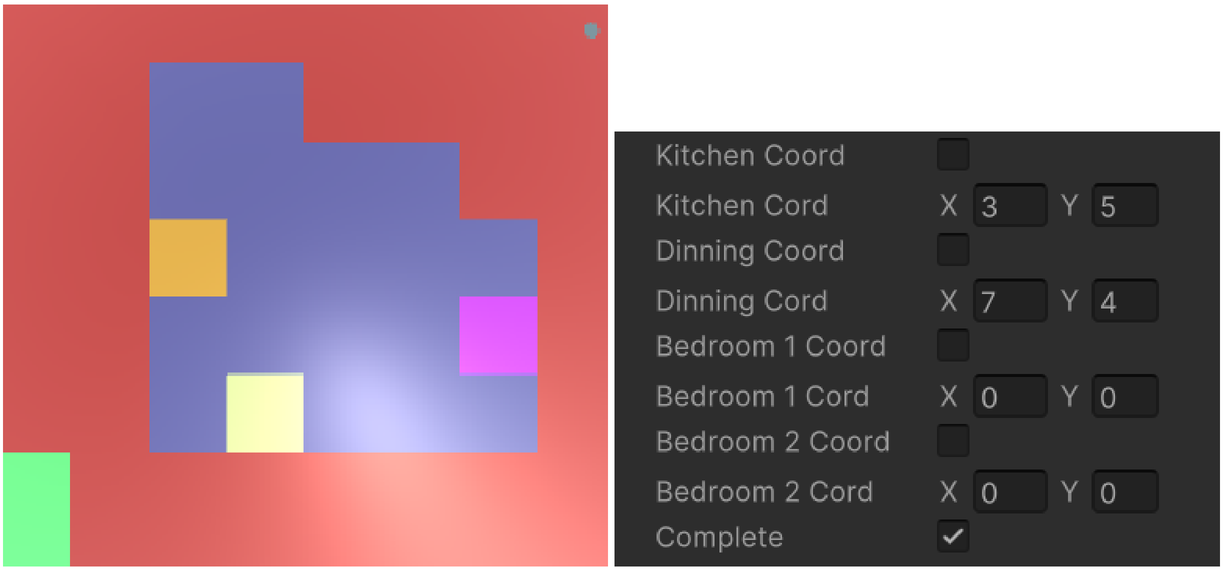


Figure 6.3: Stage 3

By completing this process a blank canvas is displayed to the users. double tapping the space , displays the 2d layout of the generated layout.Switching to the 3D layout they are given free room accessibility to roam around the models and observe the layout closer.

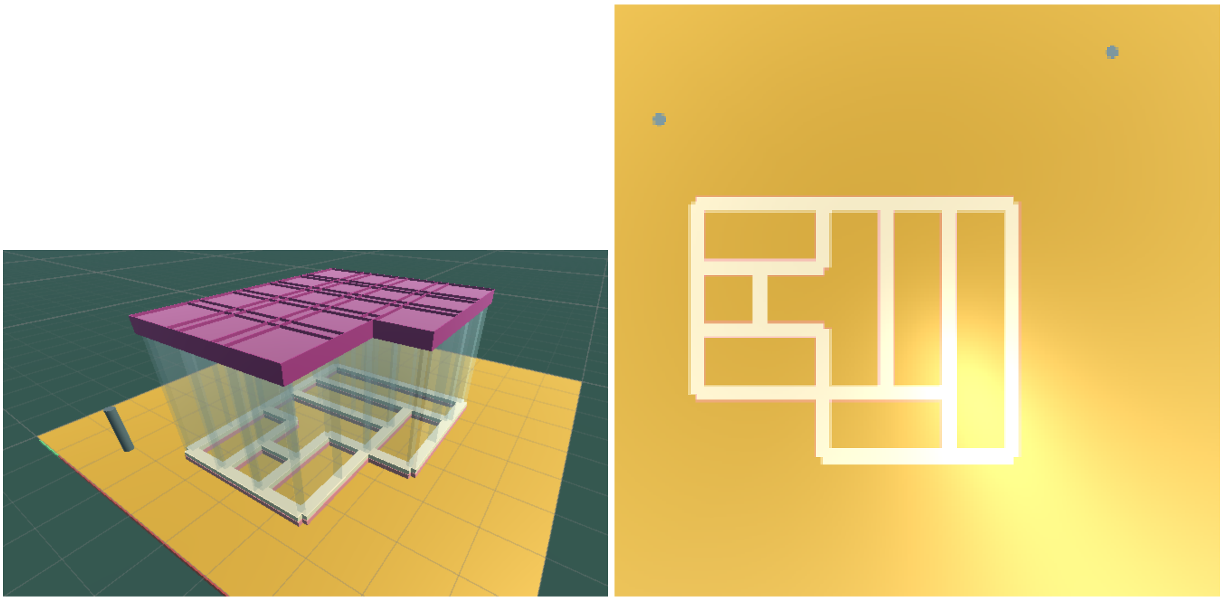


Figure 6.4: OutPut

6.2 Discussion

In addition to the current results, additional steps were taken to implement the placement of doors and windows using A* as a core component. This decision aimed to enhance the realism and functionality of the generated building layouts, allowing users to visualize the flow and accessibility of each room within the structure, as well as the integration of natural light sources. However, despite the initial intention to include these features, they were ultimately removed from the final output due to the effects of fogging, which impacted the visual appeal of the project. While the removal of these components may have resulted in a simpler visual presentation, it was deemed necessary to prioritize overall project aesthetics and user experience.Moving forward, there may be opportunities to revisit the implementation of door and window placement features, perhaps through alternative methods or optimizations to mitigate the impact of fogging.

The results and discussions showcase the effectiveness of Design EZ in simplifying residential building design. While initially considering additional features like door and window placement, the project prioritized visual appeal and user experience. Despite these omissions, Design EZ remains a highly effective platform, offering intuitive tools for layout creation. Moving forward, revisiting the implementation of such features and enhancing visual presentation will be key to ensuring its continued success in architectural design.

Chapter 7

Conclusions & Future Scope

The Design EZ project introduces a user-friendly solution for residential building design. With its intuitive interface and innovative algorithms, users can effortlessly create and customize layouts.

- Implement material decorations to enhance the visual appeal of generated layouts.
- Develop improved algorithms for layout generation to enhance efficiency and versatility.
- Introduce machine learning concepts and database integration for enhanced functionality and data management.
- Explore alternative architectural layouts to accommodate different structural designs and load-bearing mechanisms.

References

[1] L. van Aanholt and R. Bidarra, "Declarative procedural generation of architecture with semantic architectural profiles," in *2020 IEEE Conference on Games (CoG)*. IEEE, 2020, pp. 351–358.

[2] J. Zhu, H. Zhang, and Y. Wen, "A new reconstruction method for 3d buildings from 2d vector floor plan," *Computer-aided design and applications*, vol. 11, no. 6, pp. 704–714, 2014.

[3] R. Lewis and C. Séquin, "Generation of 3d building models from 2d architectural plans," *Computer-Aided Design*, vol. 30, no. 10, pp. 765–779, 1998.

[4] L. Gimenez, S. Robert, F. Suard, and K. Zreik, "Automatic reconstruction of 3d building models from scanned 2d floor plans," *Automation in Construction*, vol. 63, pp. 48–56, 2016.

[5] S. Park and H. Kim, "3dplannet: generating 3d models from 2d floor plan images using ensemble methods," *Electronics*, vol. 10, no. 22, p. 2729, 2021.

[6] J. Morisset de Pérdigo, "Floor plan generation as an optimization problem," Master's thesis, 2021.

[7] E. Karan, S. Asgari, and A. Rashidi, "A markov decision process workflow for automating interior design," *KSCE Journal of Civil Engineering*, vol. 25, no. 9, pp. 3199–3212, 2021.

Appendix A: Presentation

Design EZ

Dr. Mary Priya Sebastian
Project Guide
Asst. Professor
Dept. of CSE
RSET

Pranav Sridhar Natarajan
Rony Mons
S Gokul Raj

Contents

Problem Definition

- There are tools that only give the 2D drawings of buildings
- Miscommunication between architect and homeowner leading to dissatisfaction with the design.
- Often results in simple and repetitive layout designs that do not meet homeowner's expectations.
- Time-consuming process for creating a single design.
- Tedious task of searching for buildings with satisfactory layouts adds to the time consumption.

Project Objectives

- Develop algorithms for layout generation to automate the process of creating diverse and customized building layouts.
- Ensuring architectural stability in the building

Novelty of Idea

Unlike conventional methods, this project focuses on improving homeowners interaction in the development of the design from the start.By providing intuitive tools and interfaces, the project empowers homeowners to actively participate in shaping the design of their residential spaces, ensuring that their preferences and needs are thoroughly considered throughout the process. This early and continuous involvement of homeowners sets a new standard for residential building design, promising more personalized and satisfying outcomes for all parties involved.

Scope of Implementation

The scope is that

- It can be used by architects to quickly get a 3D model of a building
- It can be used by users who want a lot of designs in a quick span of time

Literature Review

- Floor Plan Generation as an Optimization problem
  This paper presents a method for generating diverse floor plans tailored for residential projects during their initial developmental stages. One of the principal strengths of this method lies in its simplicity and compatibility with most architectural software, rendering it highly customizable and adaptable to specific project needs.
- A Declarative Tile-Based Approach for Procedural Generation of Architecture
  The paper presents an innovative method that combines declarative and comprehensive characteristics. The approach's ease of configuration allows for the generation of diverse architectural structures, catering to various design requirements and preferences.

Literature Review

- A New Reconstruction Method for 3D Buildings from 2D Vector Floor Plan
  It offers a rapid reconstruction process, incorporating both geological and semantic information into the 3D models. Additionally, the creation of a topological model facilitates the analysis of space connectivity and relationships, providing valuable insights for architecture, urban planning, and virtual reality applications.
- Effective Interaction with Virtual Building Environments
  At the heart of the paper lies a novel method for constructing detailed frame models tai- lored to various portal types. This methodology enables the incorporation of components that cannot be accurately represented using traditional extrusion techniques, thereby enhancing the realism and fidelity of virtual building environments.

Methodology

- **Plot Dimension Input:**
  - Users provide essential input regarding the dimensions of the surface required for the project.
  - The system instantiates a corresponding grid and surface for interaction.
  - Enables the next section to start working

Methodology

- **Plot Shape Drawing:**
  - Users draw the plot shape using mouse-based raycasting.
  - The system identifies closed shapes and calculates centroids.
  - Validation ensures accurate shape drawing and closed contours.

Methodology

Algorithm 1 UserWallCreation(plainSurface)

procedure UserWallCreation(plainSurface)

  Input: plainSurface: Plain surface where the user interacts to create walls.

  Procedure:

  Create an empty array of Vector2 to store vertices of the user-created polygon.

  while Closed Polygon is not Formed **do**

1. On mouse button down event:
   (a) Perform a raycast from the mouse position to detect collision with the plain surface.
   (b) If the raycast hits the surface:
      i. Store the mouse coordinates as a Vector2 in the array.
2. On mouse button up event:
   (a) Perform a raycast from the mouse position to detect collision with the plain surface.

   If the raycast hits the surface on mouse button up:

1. Calculate the distance between the stored start and end coordinates of the mouse.
2. Determine the midpoint between the start and end coordinates.
3. Instantiate the userWall prefab at the midpoint with its scale set to half of the calculated distance.

  end while

end procedure

Methodology

- **Feasible Area Identification:**
  - Starting from the centroid of the plot shape, the system initiates a recursive function to systematically explore and identify feasible cells within the polygon.
  - The recursive function checks each cell within the polygon, determining its feasibility based on predefined criteria.
  - If a cell is deemed feasible, the function recursively explores its adjacent cells to further identify viable areas while respecting the boundaries of the polygon.

Methodology

Algorithm 2 Feasible Area Identification

```
procedure FEASIBLEAREAIDENTIFICATION(User drawn walls)
    Store the vertices of the user-drawn walls in a 2D vector array
    Calculate the centroid of the polygon formed by the user-drawn walls
    Call Flood(cell)
end procedure
procedure FLOOD(cell)
    if cell is not a wall then
        Set the cell as feasible
        Call Flood(cell(x, y + 1))
        Call Flood(cell(x, y − 1))
        Call Flood(cell(x + 1, y))
        Call Flood(cell(x − 1, y))
    end if
end procedure
```

Methodology

- **Living Room Location Selection:**
  - Users select a cell within the feasible area for the living room.
  - Interface provides visual feedback and selection assistance.
  - Centroid calculation aids in initial placement for user convenience.

Methodology

- **Optional Room Specification:**
  - Users specify areas for additional rooms like dining room, kitchen, and bedrooms.
  - Interface offers options for room selection and customization.
  - Constraints ensure user-defined areas align with feasibility considerations.

Methodology

- **Layout Generation:**
  - The system divides the feasible area into public and private sections based on user preferences.
  - Percentage ratios dictate room sizes within the layout generation algorithm.
  - Public area rooms (living, dining, kitchen) generated using a modified flooding algorithm.
  - Private area rooms (bedrooms, bathrooms) generated based on predefined constraints.
  - Iterative process ensures optimal space allocation and layout coherence.

Methodology

Algorithm 3 LayoutGeneration(feasibleAreaMatrix)

```
procedure LayoutGeneration(feasibleAreaMatrix)
    Input: feasibleAreaMatrix: Matrix representation of the feasible area including room locations.
    Output: final layout matrix
    Initialize an empty layout matrix to represent the final layout.
    for each room type starting with the living room do
        Initialize an empty array to store neighbouring cells.
        while the number of cells in the room is less than the specified percentage do
            Select a random cell from the array with allocation value set false.
            Add the selected cell to the room.
            Add neighbouring cells of the selected cell to the neighbouring array.
        end while
        Generate a random number, either 1 or 2.
        if the random number is 1 then
            Select all neighbouring cells from the array that are in the same column as the selected cell.
        else if the random number is 2 then
            Select all neighbouring cells from the array that are in the same row as the selected cell.
        end if
    end for
    set allocation of all the selected state as true.
    Add all selected neighbouring cells to the room.
    Output the final layout matrix.
end procedure
```

Methodology

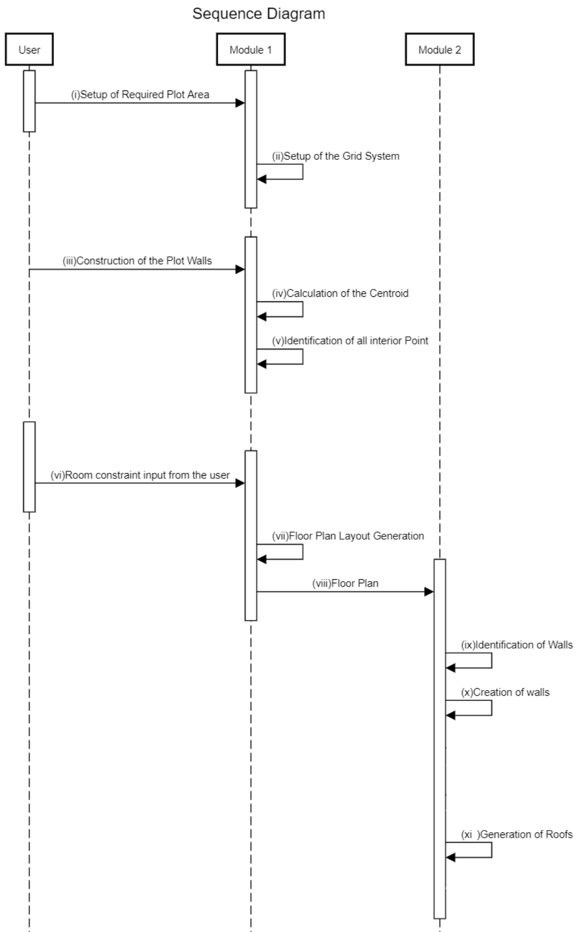
- **Wall and Roof Generation:**
  - Walls are generated between rooms and along boundaries within the feasible area.
  - Roofs are placed above feasible cells, ensuring coverage and structural integrity.
  - Algorithmic logic prevents overlapping and ensures correct placement.
  - Rendering module visualizes generated walls and roofs for user feedback.

Methodology

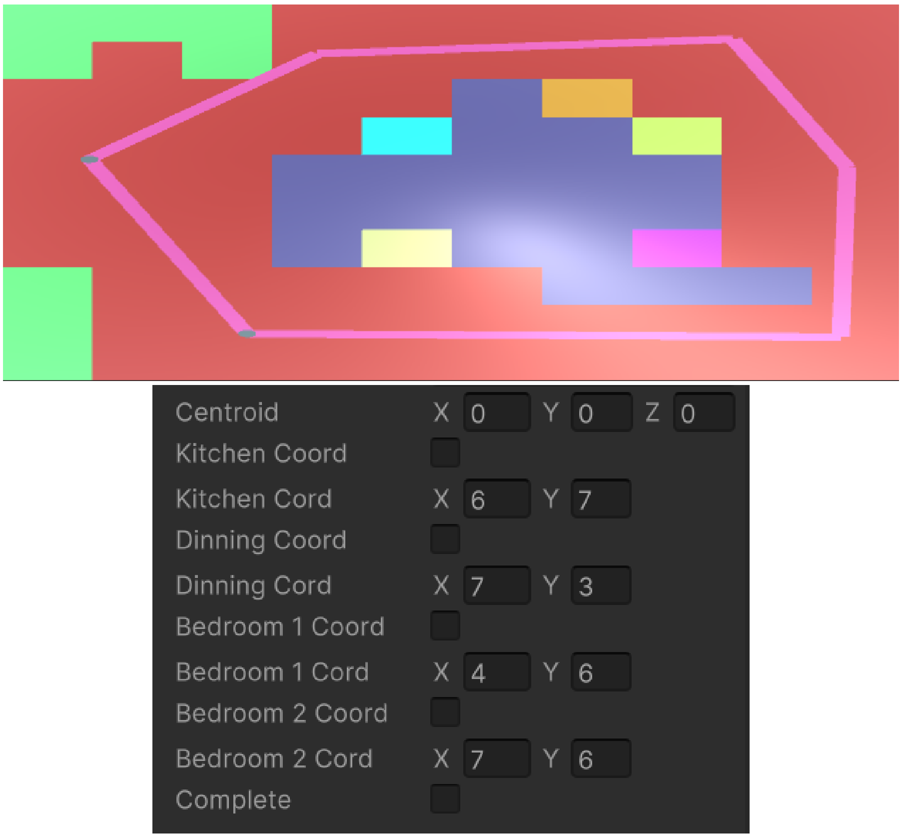
Algorithm 4 WallAndRoofGeneration(feasibleAreaMatrix)

```
procedure WallAndRoofGeneration(feasibleAreaMatrix)
    Input: feasibleAreaMatrix: Matrix representation of the land after layout generation.
    Define GameObjects:
    Define GameObjects for vertical walls, horizontal walls, and roof (1-meter square prefab).
    Repeat for Each Pair of Adjacent Cells in the Matrix:
    for each pair of adjacent cells (cell1, cell2) in the matrix do
        if cell1 and cell2 are adjacent and both are feasible cells then
            Instantiate the roof prefab at the position of cell1 with an offset of 0.
            if cell1 and cell2 represent a boundary between different rooms then
                if cell1 and cell2 are vertically adjacent then
                    Instantiate a horizontal wall prefab at the position of cell1 with an offset of 0.5.
                else if cell1 and cell2 are horizontally adjacent then
                    Instantiate a vertical wall prefab at the position of cell1 with an offset of 0.5.
                end if
            end if
        end if
    end for
    Output:
    Finalize the wall and roof generation process, providing structural elements for the building layout.
end procedure
```

Sequence Diagram

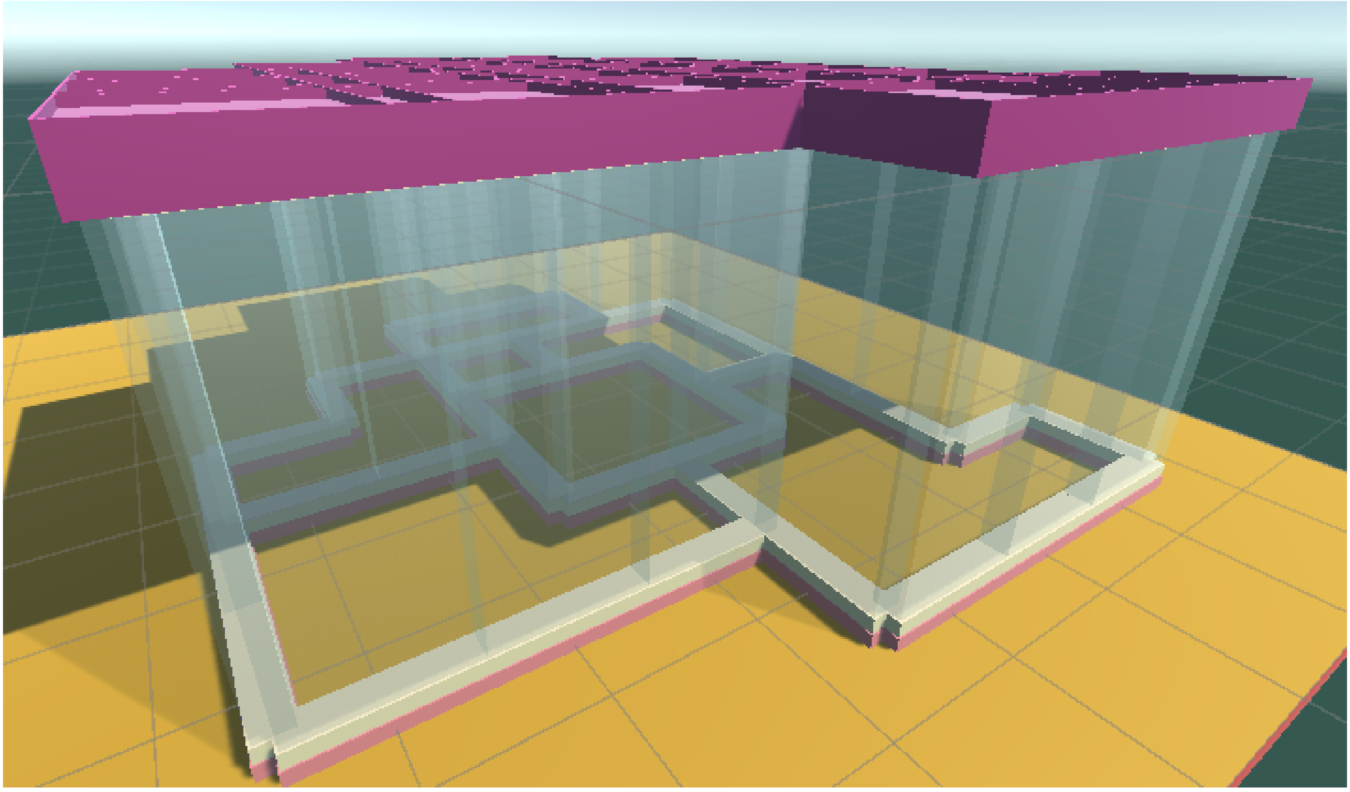


Results



User input(the locations of the rooms)

Results



Final output

Work Distribution among Team Members

Pranav Sridhar Natarajan

- Worked on reading input from the user.
- Documented the algorithms.
- Prepared the PowerPoint presentation.

Rony Mons

- Worked on the wall placement algorithm.
- Created models for the walls.
- Documented the algorithms.
- Identified conference papers.

S Gokul Raj

- Created the floor plan layout generation algorithm.
- Developed the player movement algorithm.

Conclusion

This project creates a 3D model of a residential building which is a user friendly inspiration which helps them in the process of constructing their dream house.

Future Scope

- Material decorations for the generations is not focused in this project, further development can focus in the decoration.
- Keeping this algorithm as a base and making better algorithms for the layout generation will be useful.
- Introduce the concept of ML to this project and development of database for the same
- this project is based on a simpler architectural layout where the load from the top section is transferred through the walls, different architectural layout's can be used .

References

- X. -Y. Wang, Y. -F. Liu and K. Zhang, "A Graph Grammar Approach to the Design and Validation of Floor Plans," in The Computer Journal, vol. 63, no. 1, pp. 137-150, Jan. 2020, doi: 10.1093/comjnl/bxz002
- Wu, Wenming, et al. "Data-driven interior plan generation for residential buildings." ACM Transactions on Graphics (TOG) 38.6 (2019): 1-12.
- Morisset de Pérdigo, J. R. Floor Plan Generation as an Optimization problem. MS thesis. 2021.
- L. van Aanholt and R. Bidarra, "Declarative proceduralgeneration of architecture with semantic architecturalprofiles," 2020 IEEE Conference on Games (CoG),Osaka, Japan, 2020, pp. 351-358, doi:10.1109/CoG47356.2020.9231561.
- Rick Lewis, Carlo Séquin,Generation of 3D building models from 2D architectural plans,Computer-Aided Design,Volume 30, Issue 10,1998,Pages 765-779.

Status of Paper Publication

The status of the report is as follows: the abstract has been formally submitted to the journal editor and has received approval. Presently, we are actively engaged in advancing the journal's completion, ensuring that it meets the requisite standards and effectively communicates the project's objectives and outcomes.

Appendix B: Vision, Mission, Programme Outcomes and Course Outcomes

Vision, Mission, Programme Outcomes and Course Outcomes

Institute Vision

To evolve into a premier technological institution, moulding eminent professionals with creative minds, innovative ideas and sound practical skill, and to shape a future where technology works for the enrichment of mankind.

Institute Mission

To impart state-of-the-art knowledge to individuals in various technological disciplines and to inculcate in them a high degree of social consciousness and human values, thereby enabling them to face the challenges of life with courage and conviction.

Department Vision

To become a centre of excellence in Computer Science and Engineering, moulding professionals catering to the research and professional needs of national and international organizations.

Department Mission

To inspire and nurture students, with up-to-date knowledge in Computer Science and Engineering, ethics, team spirit, leadership abilities, innovation and creativity to come out with solutions meeting societal needs.

Programme Outcomes (PO)

Engineering Graduates will be able to:

1. Engineering Knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

2. Problem analysis: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

3. Design/development of solutions: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

4. Conduct investigations of complex problems: Use research-based knowledge including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

5. Modern Tool Usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.

6. The engineer and society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal, and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

7. Environment and sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

8. Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

9. Individual and Team work: Function effectively as an individual, and as a member or leader in teams, and in multidisciplinary settings.

10. Communication: Communicate effectively with the engineering community and with society at large. Be able to comprehend and write effective reports documentation. Make effective presentations, and give and receive clear instructions.

11. Project management and finance: Demonstrate knowledge and understanding of engineering and management principles and apply these to one's own work, as a member and leader in a team. Manage projects in multidisciplinary environments.

12. Life-long learning: Recognize the need for, and have the preparation and ability to engage in independent and lifelong learning in the broadest context of technological change.

Programme Specific Outcomes (PSO)

A graduate of the Computer Science and Engineering Program will demonstrate:

PSO1: Computer Science Specific Skills
The ability to identify, analyze and design solutions for complex engineering problems in multidisciplinary areas by understanding the core principles and concepts of computer science and thereby engage in national grand challenges.

PSO2: Programming and Software Development Skills
The ability to acquire programming efficiency by designing algorithms and applying standard practices in software project development to deliver quality software products meeting the demands of the industry.

PSO3: Professional Skills
The ability to apply the fundamentals of computer science in competitive research and to develop innovative products to meet the societal needs thereby evolving as an eminent researcher and entrepreneur.

Course Outcomes (CO)

Course Outcome 1: Model and solve real world problems by applying knowledge across domains (Cognitive knowledge level: Apply).

Course Outcome 2: Develop products, processes or technologies for sustainable and socially relevant applications (Cognitive knowledge level: Apply).

Course Outcome 3: Function effectively as an individual and as a leader in diverse teams and to comprehend and execute designated tasks (Cognitive knowledge level: Apply).

Course Outcome 4: Plan and execute tasks utilizing available resources within timelines, following ethical and professional norms (Cognitive knowledge level: Apply).

Course Outcome 5: Identify technology/research gaps and propose innovative/creative solutions (Cognitive knowledge level: Analyze).

Course Outcome 6: Organize and communicate technical and scientific findings effectively in written and oral forms (Cognitive knowledge level: Apply).

++

Appendix C: CO-PO-PSO Mapping

CO-PO AND CO-PSO MAPPING

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PSO 1	PSO 2	PSO 3
CO 1	2	2	2	1	2	2	2	1	1	1	1	2	3		
CO 2	2	2	2		1	3	3	1	1		1	1		2	
CO 3									3	2	2	1			3
CO 4					2			3	2	2	3	2			3
CO 5	2	3	3	1	2							1	3		
CO 6					2			2	2	3	1	1			3

3/2/1: high/medium/low

JUSTIFICATIONS FOR CO-PO MAPPING

MAPPING	LOW/MEDIUM/ HIGH	JUSTIFICATION
100003/ CS722U.1-PO1	M	Knowledge in the area of technology for project development using various tools results in better modeling.
100003/ CS722U.1-PO2	M	Knowledge acquired in the selected area of project development can be used to identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions.

100003/ CS722U.1-PO3	M	Can use the acquired knowledge in designing solutions to complex problems.
100003/ CS722U.1-PO4	M	Can use the acquired knowledge in designing solutions to complex problems.
100003/ CS722U.1-PO5	H	Students are able to interpret, improve and redefine technical aspects for design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
100003/ CS722U.1-PO6	M	Students are able to interpret, improve and redefine technical aspects by applying contextual knowledge to assess societal, health and consequential responsibilities relevant to professional engineering practices.
100003/ CS722U.1-PO7	M	Project development based on societal and environmental context solution identification is the need for sustainable development.
100003/ CS722U.1-PO8	L	Project development should be based on professional ethics and responsibilities.
100003/ CS722U.1-PO9	L	Project development using a systematic approach based on well defined principles will result in teamwork.
100003/ CS722U.1-PO10	M	Project brings technological changes in society.

100003/ CS722U.1-PO11	H	Acquiring knowledge for project development gathers skills in design, analysis, development and implementation of algorithms.
100003/ CS722U.1-PO12	H	Knowledge for project development contributes engineering skills in computing & information gatherings.
100003/ CS722U.2-PO1	H	Knowledge acquired for project development will also include systematic planning, developing, testing and implementation in computer science solutions in various domains.
100003/ CS722U.2-PO2	H	Project design and development using a systematic approach brings knowledge in mathematics and engineering fundamentals.
100003/ CS722U.2-PO3	H	Identifying, formulating and analyzing the project results in a systematic approach.
100003/ CS722U.2-PO5	H	Systematic approach is the tip for solving complex problems in various domains.
100003/ CS722U.2-PO6	H	Systematic approach in the technical and design aspects provide valid conclusions.
100003/ CS722U.2-PO7	H	Systematic approach in the technical and design aspects demonstrate the knowledge of sustainable development.

100003/ CS722U.2-PO8	M	Identification and justification of technical aspects of project development demonstrates the need for sustainable development.
100003/ CS722U.2-PO9	H	Apply professional ethics and responsibilities in engineering practice of development.
100003/ CS722U.2-PO11	H	Systematic approach also includes effective reporting and documentation which gives clear instructions.
100003/ CS722U.2-PO12	M	Project development using a systematic approach based on well defined principles will result in better teamwork.
100003/ CS722U.3-PO9	H	Project development as a team brings the ability to engage in independent and lifelong learning.
100003/ CS722U.3-PO10	H	Identification, formulation and justification in technical aspects will be based on acquiring skills in design and development of algorithms.
100003/ CS722U.3-PO11	H	Identification, formulation and justification in technical aspects provides the betterment of life in various domains.
100003/ CS722U.3-PO12	H	Students are able to interpret, improve and redefine technical aspects with mathematics, science and

		engineering fundamentals for the solutions of complex problems.
100003/ CS722U.4-PO5	H	Students are able to interpret, improve and redefine technical aspects with identification formulation and analysis of complex problems.
100003/ CS722U.4-PO8	H	Students are able to interpret, improve and redefine technical aspects to meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
100003/ CS722U.4-PO9	H	Students are able to interpret, improve and redefine technical aspects for design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
100003/ CS722U.4-PO10	H	Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools for better products.
100003/ CS722U.4-PO11	M	Students are able to interpret, improve and redefine technical aspects by applying contextual knowledge to assess societal, health and consequential responsibilities relevant to professional engineering practices.
100003/ CS722U.4-PO12	H	Students are able to interpret, improve and redefine technical aspects for demonstrating the knowledge of, and need for sustainable development.

100003/ CS722U.5-PO1	H	Students are able to interpret, improve and redefine technical aspects, apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
100003/ CS722U.5-PO2	M	Students are able to interpret, improve and redefine technical aspects, communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
100003/ CS722U.5-PO3	H	Students are able to interpret, improve and redefine technical aspects to demonstrate knowledge and understanding of the engineering and management principle in multidisciplinary environments.
100003/ CS722U.5-PO4	H	Students are able to interpret, improve and redefine technical aspects, recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.
100003/ CS722U.5-PO5	M	Students are able to interpret, improve and redefine technical aspects in acquiring skills to design, analyze and develop algorithms and implement those using high-level programming languages.
100003/ CS722U.5-PO12	M	Students are able to interpret, improve and redefine technical aspects and contribute their engineering skills

		in computing and information engineering domains like network design and administration, database design and knowledge engineering.
100003/ CS722U.6-PO5	M	Students are able to interpret, improve and redefine technical aspects and develop strong skills in systematic planning, developing, testing, implementing and providing IT solutions for different domains which helps in the betterment of life.
100003/ CS722U.6-PO8	H	Students will be able to associate with a team as an effective team player for the development of technical projects by applying the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
100003/ CS722U.6-PO9	H	Students will be able to associate with a team as an effective team player to Identify, formulate, review research literature, and analyze complex engineering problems
100003/ CS722U.6-PO10	M	Students will be able to associate with a team as an effective team player for designing solutions to complex engineering problems and design system components.
100003/ CS722U.6-PO11	M	Students will be able to associate with a team as an effective team player, use research-based knowledge and research methods including design of experiments, analysis and interpretation of data.

100003/ CS722U.6-PO12	H	Students will be able to associate with a team as an effective team player, applying ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
100003/ CS722U.1-PSO1	H	Students are able to develop Computer Science Specific Skills by modeling and solving problems.
100003/ CS722U.2-PSO2	M	Developing products, processes or technologies for sustainable and socially relevant applications can promote Programming and Software Development Skills.
100003/ CS722U.3-PSO3	H	Working in a team can result in the effective development of Professional Skills.
100003/ CS722U.4-PSO3	H	Planning and scheduling can result in the effective development of Professional Skills.
100003/ CS722U.5-PSO1	H	Students are able to develop Computer Science Specific Skills by creating innovative solutions to problems.
100003/ CS722U.6-PSO3	H	Organizing and communicating technical and scientific findings can help in the effective development of Professional Skills.