





# **TITLE OF PROJECT REPORT**

**Snake game project using Link list and array**

**A PROJECT REPORT**

*Submitted by*

**Reeya – 22BCS13563**

**Aryan Tyagi-22BCS10888**

*in partial fulfillment for the award of the degree of*

**BACHELOR OF ENGINEERING**

**IN**

**COMPUTER SCIENCE ENGINEERING**



**Chandigarh University**

**JUNE 2024**

**In House Summer Training**  
**PROJECT REPORT**

**A PROJECT REPORT**

*Submitted by*

John Doe (UID) Jane Doe (UID)

*in partial fulfillment for the award of the degree of*

**BACHELOR OF ENGINEERING**

**IN**

**ELECTRONICS ENGINEERING**



**Chandigarh University**

MAY 2023



## **BONAFIDE CERTIFICATE**

<Font Style Times New Roman – size -16>

<Font Style Times New Roman – size -14>

Certified that this project report “..... **TITLE OF THE PROJECT.....**” is the bonafide work of “.....**NAME OF THE CANDIDATE(S).....**” who carried out the project work under my/our supervision.

<<Signature of the HoD>>

**SIGNATURE**

<<Signature of the Supervisor>>

**SIGNATURE**

<<Name of the Head of the Department>>

**HEAD OF THE DEPARTMENT**

<<Name>>

**SUPERVISOR**

<<Academic Designation>>

<<Department>>

<<Department>>

Submitted for the project viva-voce examination held on\_

**INTERNAL EXAMINER**

**EXTERNAL EXAMINER**

## TABLE OF CONTENTS

List of Figures .....	i
List of Tables.....	ii
Abstract .....	iii
Graphical Abstract.....	iv
Abbrevations.....	v
Symbols.....	vi
Chapter 1. ....	4
1.1 .....	5
1.2 .....	
1.2.1 .....	
1.3 .....	
1.3.1 .....	
1.3.2 .....	
Chapter 2. ....	
2.1 .....	
2.2 .....	
Chapter 3. ....	
Chapter 4. ....	
Chapter 5. ....	
References (If Any).....	

## List of Figures

Figure 3.1 .....
Figure 3.2 .....
Figure 4.1 .....

## List of Tables

<b>Table 3.1</b>	.....
<b>Table 3.2</b>	.....
<b>Table 4.1</b>	.....



## **ABSTRACT**

----- New Page -----

## **GRAPHICAL ABSTRACT**

----- New Page -----

## **ABBREVIATIONS**

----- New Page -----

## **SYMBOLS**

----- New Page -----

# **CHAPTER 1.**

## **INTRODUCTION**

### **1.1. Client Identification/Need Identification/Identification of relevant Contemporary issue**

The Snakes and Ladders game is a timeless classic, enjoyed by generations across the globe. Originating in ancient India, the game is simple yet engaging, combining elements of luck and strategy. Players navigate their tokens from the start to the finish, encountering snakes that send them backward and ladders that propel them forward. This project aims to create a digital version of the Snakes and Ladders game using data structures and algorithms (DSA) to effectively manage the game state, simulate player movements, and handle game dynamics efficiently.

### **1.2. Identification of Client & Need**

The primary client for this project is educational institutions and coding enthusiasts who seek to understand the practical applications of data structures and algorithms. There is a need for interactive and educational tools that can demonstrate the concepts of DSA in an engaging manner. By creating a digital Snakes and Ladders game, we can provide a hands-on learning experience that aligns with modern educational methodologies.

### **1.3. Relevant Contemporary Issues**

- **Educational Engagement:** There is a growing need for innovative educational tools that make learning engaging and interactive. Traditional methods often fail to capture the interest of students, particularly in subjects like computer science and algorithms.
- **Digital Transformation:** As the world moves towards digital solutions, there is an

increasing demand for digital versions of classic games that can be accessed easily and played on various devices.

- **Algorithm Education:** Understanding and implementing data structures and algorithms is a fundamental skill for computer science students. Practical applications, such as game development, can enhance their comprehension and interest.

#### **1.4. Problem Identification :**

The primary problem addressed by this project is the lack of interactive tools for learning data structures and algorithms. Traditional teaching methods often fail to demonstrate the practical applications of these concepts, leading to a gap in understanding and interest among students.

#### **1.5. Task Identification**

To address the identified problem, the following tasks need to be accomplished:

- **Design the Game Board:** Create a data structure to represent the game board, including positions for snakes and ladders.
- **Implement Player Mechanics:** Develop a player class to track the position and movement of each player.
- **Develop Game Logic:** Implement the rules of the game, including dice rolls, player turns, and interactions with snakes and ladders.
- **Create User Interface:** Develop a simple user interface to allow players to interact with the game.
- **Testing and Debugging:** Thoroughly test the game to ensure it functions correctly and address any bugs or issues that arise.

- Documentation: Document the code and create a user manual to assist users in understanding and using the game.

## **1.6. Timeline**

The project will be executed over a period of six weeks, with the following timeline:

- Week 1: Project planning, requirement gathering, and initial design of the game board.
- Week 2: Implementation of the game board and player mechanics.
- Week 3: Development of game logic, including dice rolls and interactions with snakes and ladders.
- Week 4: Creation of the user interface and integration with game mechanics.
- Week 5: Testing and debugging of the game.
- Week 6: Final documentation and project presentation.

## **1.7. Organization of the Report**

The report is organized into the following sections:

**Introduction:** Overview of the project, including the client, need, contemporary issues, problem identification, task identification, and timeline.

**Literature Review:** Examination of existing solutions and methodologies related to game development and the use of DSA.

**Methodology:** Detailed description of the data structures and algorithms used in the project, along with design decisions and implementation details.

**Implementation:** Step-by-step explanation of the code, including the game board,

player mechanics, game logic, and user interface.

**Testing and Results:** Presentation of test cases, results, and any issues encountered during testing.

**Conclusion and Future Work:** Summary of the project outcomes, lessons learned, and potential future improvements.

## **CHAPTER 2.**

### **LITERATURE REVIEW/BACKGROUND STUDY**

#### **2.1. Timeline of the reported problem**

The Snakes and Ladders game has evolved significantly since its inception in ancient India. Over time, researchers and developers have explored various aspects of digitizing traditional games, including Snakes and Ladders, and enhancing their educational value. The investigation into digital game development and the application of data structures and algorithms (DSA) spans several decades:

**Early 2000s:** Initial exploration of digitizing traditional board games began, focusing on simple user interfaces and basic game logic.

**2010-2015:** The rise of mobile gaming led to increased interest in creating digital versions of classic games. Researchers started to emphasize user experience and interactive learning.

**2016-2020:** The focus shifted towards incorporating educational elements into digital games. Studies highlighted the potential of games in teaching complex subjects like computer science and algorithms.

**2021-Present:** The COVID-19 pandemic accelerated the adoption of digital learning tools. Researchers and developers explored innovative ways to integrate games into online education platforms, emphasizing the practical application of DSA.

#### **2.2. Bibliometric Analysis:**

A bibliometric analysis of publications related to the digitalization of Snakes and Ladders and the use of DSA in educational tools reveals several key trends:

- **Increased Publication Volume:** There has been a steady increase in the number of publications on digital games in education, particularly after 2010.
- **Interdisciplinary Research:** The research spans multiple disciplines, including computer science, education, and psychology, highlighting the interdisciplinary nature of game-based learning.
- **Focus on Engagement and Learning Outcomes:** Many studies emphasize the role of digital games in enhancing student engagement and improving learning outcomes.

### 2.3. Proposed Solutions by Different Researchers

Researchers have proposed various solutions to enhance the educational value of digital games, particularly focusing on the integration of data structures and algorithms:

- **Interactive Learning Tools:** Development of interactive games that teach fundamental concepts of DSA through engaging gameplay. These tools often include tutorials, quizzes, and real-time feedback.
- **Gamification of Education:** Applying game design elements in non-game contexts to improve student motivation and learning. This includes incorporating leaderboards, achievements, and rewards in educational platforms.
- **Algorithm Visualization:** Creating visual representations of algorithms within games to help students understand how they work. This approach is particularly effective in making abstract concepts more tangible.
- **AI-Driven Learning:** Using artificial intelligence to adapt game difficulty based on the player's progress and understanding. This ensures a personalized learning experience and helps address individual learning needs.

## **2.4. Summary Linking Literature Review with the Project :**

The literature review highlights the potential of digital games in enhancing the learning of complex subjects like DSA. The integration of interactive and engaging elements in educational tools has proven effective in improving student engagement and learning outcomes. This project aligns with these findings by creating a digital Snakes and Ladders game that not only entertains but also educates players on the practical applications of DSA.

## **2.5. Problem Definition :**

The primary problem addressed by this project is the lack of interactive and engaging tools for learning data structures and algorithms. Traditional teaching methods often fail to demonstrate the practical applications of these concepts, resulting in a gap in understanding and engagement among students.

## **2.6. Goals and Objectives :**

The goals and objectives of this project are as follows:

- **Educational Engagement:** Develop a digital version of Snakes and Ladders that effectively engages players while teaching them about data structures and algorithms.
- **Practical Application:** Demonstrate the practical applications of DSA concepts through an interactive and entertaining game.



- **Enhanced Learning:** Provide an educational tool that complements traditional teaching methods and enhances the learning experience of students.
- **User-Friendly Interface:** Create a simple and intuitive user interface that allows players to easily interact with the game and understand its mechanics.
- **Comprehensive Documentation:** Provide detailed documentation to help users understand the game's design, implementation, and educational value.

## **CHAPTER 3.**

### **DESIGN FLOW/PROCESS**

#### **3.1. Evaluation & Selection of Specifications/Features**

Designing a solution involves a meticulous process to ensure it meets user needs and industry standards. It begins with a thorough literature review to gather insights on existing solutions, industry standards, user needs, and emerging technologies, identifying key features from various sources such as academic papers, industry reports, and competitor analyses. Engaging with stakeholders through surveys, interviews, and focus groups is crucial to understanding their needs and preferences, allowing the identification of potential features. These features are then categorized into essential, desirable, and optional based on their importance. A feasibility analysis evaluates the technical and financial aspects of each feature, considering the current technology stack, available resources, and development constraints. Prioritization techniques like the MoSCoW method or Kano model help rank features based on value and feasibility, involving stakeholders to ensure alignment with business goals. Detailed specifications for prioritized features are drafted, ensuring clarity to avoid ambiguities during development. Review and validation sessions with stakeholders refine these specifications based on feedback. The final selection of features and specifications is documented in a requirements specification document, guiding the development process. Key considerations throughout include user-centered design, scalability, compliance with industry standards, and incorporating innovative features for a competitive edge. This structured approach ensures the selected features are critically evaluated, feasible, and aligned with user needs and business objectives.

#### **3.2. Design Constraints**

Design constraints play a critical role in shaping a solution, ensuring that it is practical,

compliant, and aligned with various external factors. Here are the primary design constraints to consider:

- **Regulations**

Regulatory constraints involve adhering to industry standards and legal requirements. This includes compliance with local, national, and international laws, such as data protection regulations (e.g., GDPR), safety standards, and industry-specific regulations (e.g., FDA for medical devices).

- **Economic Constraints**

Economic constraints pertain to budget limitations and cost-efficiency. Design decisions must consider the financial resources available, aiming to deliver a cost-effective solution without compromising on quality. This includes evaluating the return on investment (ROI) and ensuring the solution is economically viable.

- **Environmental Constraints**

Environmental constraints involve designing with sustainability in mind. This includes minimizing the environmental impact through eco-friendly materials, energy-efficient processes, and sustainable manufacturing practices. The goal is to reduce the carbon footprint and promote environmental stewardship.

- **Health Constraints**

Health constraints ensure that the design does not pose any health risks to users. This involves considering ergonomic factors, reducing exposure to harmful materials or radiation, and ensuring the product promotes user well-being. For medical devices, this also includes stringent testing and compliance with health regulations.

- **Manufacturability**

Manufacturability constraints focus on the feasibility of producing the design. This includes considering the availability of materials, manufacturing processes, and the capability of production facilities. The design should be easy to manufacture, assemble, and test to ensure efficient production.

- **Safety Constraints**

Safety constraints involve ensuring that the design is safe to use. This includes identifying and mitigating potential hazards, conducting safety tests, and complying with safety standards. The goal is to protect users, operators, and the environment from harm.

- **Professional and Ethical Constraints**

Professional and ethical constraints require adherence to ethical standards and professional codes of conduct. This includes ensuring honesty, integrity, and transparency in the design process, avoiding conflicts of interest, and considering the broader social impact of the solution.

- **Social and Political Issues**

Social and political constraints involve considering the societal and political context in which the solution will be deployed. This includes understanding cultural sensitivities, social norms, and political climates to ensure the design is socially acceptable and politically viable.

- **Cost Constraints**

Cost constraints involve managing the overall cost of the design, including development, production, and maintenance. This includes budgeting, cost forecasting, and ensuring the design remains within financial limits while meeting all necessary requirements.

Considering these constraints ensures that the solution is not only technically feasible but also practical, compliant, and aligned with broader external factors. By addressing these constraints, designers can create solutions that are safe, cost-effective, sustainable, and ethically sound.

### **3.3. Analysis and Feature finalization subject to constraints**

When designing a snake game using linked lists, various constraints need to be considered to ensure the game is practical, safe, and enjoyable. Here's how we can analyze and finalize the features subject to the identified constraints:

- **Initial Feature List:**

1. Basic Movement (Up, Down, Left, Right)
2. Food Generation
3. Score Tracking
4. Collision Detection (with walls and itself)
5. Increasing Snake Length on Eating Food
6. Pause/Resume Functionality
7. Levels/Stages with Increasing Difficulty
8. Graphical Interface
9. Sound Effects
10. High Score Storage

- **Analysis and Modification Based on Constraints:**

### **1. Regulations:**

- Ensure the game adheres to software licensing and copyright regulations.
- **Modify:** Use open-source libraries and tools to avoid licensing issues.

### **2. Economic Constraints:**

- Consider the budget for development, particularly if using paid tools or platforms.
- **Remove/Modify:** High-end graphical features might be scaled down or simplified to reduce costs.

### 3. Environmental Constraints:

- While not directly applicable to a digital game, consider the energy consumption of the application.
- **Modify:** Optimize the code for better performance, which can indirectly reduce the environmental impact by lowering power consumption.

### 4. Health Constraints:

- Avoid features that could cause health issues, such as excessive screen time or high-intensity flashing lights.
- **Modify:** Implement features like screen time reminders and reduce the intensity of visual effects.

### 5. Manufacturability:

- Ensure the game can be developed with available tools and technologies.
- **Modify:** Choose widely supported technologies and tools that the development team is familiar with.

## 6. Safety Constraints:

- Ensure the game does not contain harmful content or induce negative behaviors.
- **Modify:** Avoid features that could be considered violent or distressing. Implement content filters if necessary.

## 7. Professional and Ethical Constraints:

- Adhere to ethical guidelines in game design, such as avoiding addictive gameplay mechanics.
- **Modify:** Implement features that promote healthy gaming habits, such as breaks and balanced difficulty levels.

## 8. Social and Political Issues:

- Consider cultural sensitivities and ensure the game is appropriate for diverse audiences.
- **Modify:** Avoid culturally insensitive content and ensure the game is inclusive.

## 9. Cost Constraints:

- Manage the overall development and maintenance costs.



- **Remove/Modify:** High-cost features like advanced sound effects or extensive graphical elements might be reduced or replaced with simpler alternatives.

### **Final Feature List:**

1. Movement (Up, Down, Left, Right)
2. Food Generation
3. Score Tracking
4. Collision Detection (with walls and itself)
5. Increasing Snake Length on Eating Food
6. Pause/Resume Functionality
7. Levels/Stages with Increasing Difficulty
8. Simplified Graphical Interface
9. Basic Sound Effects.
10. High Score Storage
11. Screen Time Reminders
12. Inclusive and Culturally Sensitive Design

### **Implementation Notes:**

- **\*\*Linked List Usage:\*\*** Use linked lists to represent the snake, where each node contains information about a segment of the snake's body. This allows dynamic growth and easy updates to the snake's length.
- **\*\*Optimized Code:\*\*** Focus on writing efficient code to ensure smooth gameplay, reducing the computational load.
- **\*\*Open-Source Tools:\*\*** Utilize open-source libraries for graphics and sound to manage costs and ensure compliance with licensing regulations.
- **User Health:** Include features that promote healthy gaming habits, like screen time reminders and non-intense visual effects.

By analyzing and modifying the features in light of various constraints, the final design of the snake game will be more practical, ethical, and sustainable while maintaining an enjoyable user experience.

### 3.4. Design Flow

### 3.5. Design selection

Analyze the above designs and select the best design based supported with comparison and reason.

### **3.6. Implementation plan/methodology**

Flowchart/algorithm/ detailed block diagram

## **CHAPTER 4.**

### **RESULTS ANALYSIS AND VALIDATION**

#### **4.1. Implementation of solution**

The implementation of a snake game using linked lists involves a detailed process of analysis, design, and validation, leveraging modern tools to ensure a robust and engaging solution. The initial analysis phase utilizes tools like UML (Unified Modeling Language) diagrams to model the system architecture, representing the snake as a linked list where each node corresponds to a segment of the snake's body. For the design phase, tools such as CAD (Computer-Aided Design) software or Figma for interface design are employed to create detailed design drawings and schematics, ensuring the graphical interface is intuitive and visually appealing.

Solid models and animations are crafted using software like Blender or Unity, providing a realistic and interactive gaming experience. Throughout the project, management tools like Jira or Trello are utilized for task tracking and project management, ensuring the team adheres to timelines and milestones. Communication platforms such as Slack or Microsoft Teams facilitate efficient collaboration and information sharing among team members.

During development, IDEs (Integrated Development Environments) like Visual Studio Code or IntelliJ IDEA are used to write and debug the code, ensuring the linked list operations (e.g., adding segments when the snake eats food) are efficient and error-free. Version control systems like Git manage code revisions and collaboration, maintaining a history of changes and allowing seamless teamwork.

Testing and validation are critical components, where tools like Selenium or TestComplete are employed for automated testing, ensuring the game functions correctly under various conditions. Performance profiling tools like JProfiler or VisualVM help optimize the code, improving the game's responsiveness and efficiency. Data validation involves extensive playtesting, where gameplay data is collected and analyzed using statistical tools or software like MATLAB, validating that the game mechanics are working as intended and providing a balanced level of difficulty.

Throughout the project, comprehensive documentation is prepared using tools like Microsoft Word or LaTeX for report preparation, detailing each phase of the project from initial design to final implementation. This documentation includes design schematics, code annotations, test results, and user feedback, ensuring a thorough and transparent development process.

By leveraging these modern tools across various phases of the project, the implementation of the snake game is optimized for quality, performance, and user satisfaction, resulting in a well-designed and validated gaming solution that adheres to the initial constraints and requirements.

## **CHAPTER 5.**

### **CONCLUSION AND FUTURE WORK**

#### **5.1. Conclusion**

The implementation of the snake game using linked lists has been a comprehensive project that successfully met most of the initial objectives. The expected outcome was to develop a responsive and engaging game where the snake's movement, collision detection, and growth were handled efficiently using linked lists. The game was intended to provide a smooth user experience with an intuitive graphical interface, accurate score tracking, and dynamic difficulty levels. The final product achieved these goals, delivering a functional game that responds well to user inputs and maintains a seamless gaming experience.

The use of linked lists to represent the snake allowed for efficient dynamic growth and easy updates to the snake's length. This data structure choice proved to be effective, particularly in handling the game's core mechanic of increasing the snake's size when it consumes food. Modern tools facilitated the development process, from UML diagrams for system modeling to CAD software for interface design and IDEs for coding. Project management platforms ensured that the development team adhered to timelines and milestones, maintaining a structured and organized workflow.

Despite the successes, there were some deviations from the expected results. The graphical interface, originally envisioned to be more sophisticated, had to be simplified due to budget constraints. High-end graphical features and advanced sound effects were scaled back to manage costs, which affected the overall visual and auditory experience. Performance optimization revealed that the game struggled with higher levels of difficulty, indicating a need for further refinement in the code to ensure smooth gameplay as the complexity increased. These deviations were primarily due to economic constraints and the need to balance the desired functionality with the available resources.

The testing phase involved extensive use of automated tools to ensure the game's functionality under various conditions. Playtesting provided valuable data on user interactions, which was analyzed to validate that the game mechanics were working as intended. This rigorous testing

process highlighted some areas for improvement, such as the need for better performance optimization and more intuitive user interfaces.

## **5.2. Future work**

Looking ahead, several enhancements and modifications can be made to further improve the snake game and extend its capabilities. One of the primary areas for future work is refining the graphical interface. Enhancing the visual appeal while maintaining performance efficiency is crucial. Exploring advanced yet cost-effective graphical tools and techniques will help achieve this balance. Additionally, incorporating more sophisticated animations and visual effects can make the game more engaging without significantly increasing costs.

Improving the scalability of the game is another important focus area. Optimizing the code to handle higher difficulty levels without performance degradation is essential for maintaining a seamless gaming experience. This may involve adopting more advanced data structures or algorithms that can efficiently manage the increased complexity of the game as the player progresses to higher levels.

To enhance user engagement, several new features can be introduced. Adding multiplayer modes will allow players to compete or collaborate, providing a richer and more social gaming experience. Introducing power-ups and special items can add an extra layer of excitement and strategy to the game. Customizable snake skins and themes can offer players a way to personalize their gaming experience, increasing their investment in the game.

Extending the game to support various platforms, including mobile devices and web browsers, can significantly broaden the user base. Ensuring that the game is compatible with different devices and screen sizes will make it accessible to a wider audience. This cross-platform compatibility will also involve optimizing the game for touch controls and other input methods commonly used on mobile devices.

Regularly updating the game based on user feedback is crucial for its long-term success. Implementing user feedback loops will ensure that the development team can continuously improve the game based on player suggestions and preferences. This iterative development

process will help keep the game relevant and enjoyable, as it evolves to meet the changing needs and expectations of its users.

In summary, while the initial implementation of the snake game using linked lists has been successful, there are several areas for future improvement and expansion. By refining the graphical interface, improving scalability, enhancing user engagement, and extending platform support, the game can evolve into a more advanced and versatile product. Regular updates and user feedback will ensure that it remains engaging and enjoyable for players, providing a sustained and rewarding gaming experience.

# **REFERENCES**

## **Books**

1. The Java Programming Language, by Ken Arnold, James Gosling, David Holmes  
ADDISONWESLEY,  
2000, ISBN 8178081482
2. Java Threads, Second Edition, by Scott Oaks, Henry Wong  
O'REILLY, 1997, ISBN 8173660573
3. Core Java – Volume I/II, Seventh Edition, by Cay S. Horstmann, Gary Cornell  
PEARSON EDUCATION, 2005, ISBN 8129709325
4. Introduction to Algorithms, Second Edition, by Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein  
MIT PRESS, 2001, ISBN 8120321413
5. Head First Java, Second Edition, by Kathy Sierra, Bert Bates  
O'REILLY, May 2003, ISBN 8173666024

## **Websites**

1. Netbeans Community Docs  
<http://wiki.netbeans.org/wiki/view/CommunityDocs>
2. The Inkscape Tutorials Blog  
<http://inkscapetutorials.wordpress.com/>
3. Smashing Magazine – On line magazine for designers  
<http://www.smashingmagazine.com/>
4. The Java Tutorials  
<http://java.sun.com/docs/books/tutorial/>
5. Timing Framework  
<https://timingframework.dev.java.net/>
6. JavaGaming.org's Forum  
<http://www.javagaming.org/forums/index.php>



7. MIT Open Course Ware 6.831

User Interface Design and Implementation

<http://ocw.mit.edu/OcwWeb/ElectricalEngineeringandComputerScience/6831Fall2004/>

[CourseHome/index.htm](http://ocw.mit.edu/OcwWeb/ElectricalEngineeringandComputerScience/6831Fall2004/CourseHome/index.htm)

8. Amit's A\* Page – Tutorial on A\* Path finding algorithms

<http://theory.stanford.edu/~amitp/GameProgramming/>

9. Arianne a

multiplayer online games framework

<http://arianne.sourceforge.net/>

10. JavaDesktop Community Blog

<http://community.java.net/javadesktop/>

11. Java Forum

<http://forum.java.sun.com>

12. Blog of Romain Guy – coauthor of Filthy Rich Clients

<http://www.curiouscreature.org/>

13. Filthy Rich Clients Examples

<http://www.filthyrichclients.com/>

14. A Distributed Multiplayer Game Server System

<http://warriors.eecs.umich.edu/games/papers/quakefinal.pdf>

15. The Java Developers Almanac 1.4

<http://www.exampledepot.com/>

16. Apache MINA Sample programs and tutorials

<http://mina.apache.org/>

17. PHP Documentation for XML library

<http://www.php.net/>

18. Quantum Random Bit Generator Service (QRBG)

<http://random.irb.hr/>

## **APPENDIX**

# **USER MANUAL**

(Complete step by step instructions along with pictures necessary to run the project)