

Rubick Cube Game Project

A SYNOPSIS on

Major Project (BCA-605)

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

BACHELOR OF COMPUTER APPLICATION

SUBMITTED BY

Name: Prakash Sahu

University Roll No.: 200845106052

Batch (2020-23)

Session (Even, 2022-23)



Ch. Charan Singh University, Meerut
Uttar Pradesh, INDIA

Under the Supervision of

Mr./Dr. Ashish Bhatnagar

Associate / Assistant Professor



S.NO**INDEX****Page no.**

1	Acknowledgement	
2	Introduction	
3	Problem Statement	
4	Game Development lifecycle	
5	Game Requirement Specifications	
6	Software/Hardware/Technical Requirements	
7	Methodology	
8	Cube String Notation Diagram	
9	Cube Map Implementation and Cube Facelet Diagram	
10	References	

ACKNOWLEDGMENT

I am very grateful to my major project (605) semester 6 mentor Mr./Dr. Ashish Bhatnagar, for giving his valuable time and constructive guidance in preparing this Synopsis and Project (605). It would not have been possible to complete this Project (605) in the short period of time without his kind encouragement and valuable guidance.

DATE:

SIGNATURE:

INTRODUCTION TO TOPIC

This, “Rubick” cube project aims to make a Rubic Cube game using Unity3D Game engine for demonstration of Technology learned by students of BCA Semester 6.

This is a Simple game which aims to show how basic technologies can be used to create a simple and real-life like games and entertainment. For this, we need a game that is user-friendly,
Simple and eye-catching

This game will be made using basic technologies such C#, Mono, .NET Core 7, Unity engine and Unity’s own API for Gamplay programming.

PROBLEM STATEMENT

In this Problem statement we are going to describe What you are going to design/Develop/Study, as required by format provided to us by our project mentor.

We will be making a basic simple rubic cube game which we (Team-members) have collectively agreed to call "Rubick". This game provides an easy and portable way to play the game anywhere and everywhere to a large user base. However, there is a lot of challenge for creating a game which let the user relax and not be a annoyance for them. Today most of the games commonly on Google Playstore and Apple Appstore or Steam are riddled with ads and in-app purchase schemes. To counter these, we have kept the game completely ad-free with no tracker, analytics, telemetry and above all it is Open-Sourced, And will remain so.

So make this a pleasing experience for players, we aim to make this game have a mixture of fluid motion, which is soothing and play.

Software Development Lifecycle

Game Development Lifecycle



(GRS)

(Game Requirement Specifications)

Introduction

The purpose of this document is to define and describe the requirements of the project and to spell out the system's functionality and its constraints.

General Description

1. Product Functions

The product should make entire gaming experience easier and streamlined for the player and entertaining by simplifying the UI using, Usability and user experience methodology.

2. User Objectives

The user wants a game that will be fluid and easy to use. The game must facilitate the good performance and ease of playing.

3. General Constraints

Constraints include an easy to use interface for the game, a Windows Platform or, a Mac or Linux Distro. Also, it must be Fast and ad-free, and easy to navigate and use.

Functional Requirements

functions provided on Rubicks shall be easily understandable.

1. Game shall be able to run on the laptop machine or server.
2. Very less bug while maintaing performance.
3. Limited network / Wi-Fi availability shouldn't affect the game.

4. The above stated factor is a risk we have encountered. Eliminate it by reducing the dependency of our game on internet.
5. users should be able to spin the cube in various direction to play effectively
6. user can rotate the cube sides.
7. User can shuffle the cube to get a random state of cube.
8. This requirement is the basis of the project; all other aspects depend on it.

Interface Requirements

1. User Interfaces

- GUI – The user interface for this game is the interface is simple and let the player dive in directly into the game without going through a traditional main menu style that the general games uses. It include a game cube, cube-map, Shuffle functionality button, Solve functionality using kociemba Algorithem in less than 22 moves in total,
- CLI – There is no command line interface
- API – There is no API for the game

However the game is available for 3 platforms, namely Windows, MacOS and all Linux Distros having native x86_64 architecture. making it cross-platform

2. Hardware Interfaces

The game uses the GitHub for hosting the game files during development. The game will also be released on GitHub Release v1.0

3. Software Interfaces

The Game's interface can be accessed by using a modern computer or a smartphone device through WebGL(Use Firefox) although is designed especially for Desktop PCs.

Cost Feasibility

There is no particular budget for game as the project is done by some college students. And it is a known fact that college students are always broke and we also don't have any job.

Software Requirements

1. Development Requirements.

- Any Modern OS (Linux, Windows or MacOS)
- Visual Studio Code/Codium or Visual Studio
- Git and GitHub
- Unity Engine
- Blender
- Photoshop

2. Consumer Requirements.

- Any modern Web browser (Microsoft Edge, Google Chrome etc.)
- Any Modern OS (Linux, Windows or MacOS)
- Active Internet Connection (For Download only)

Hardware Requirements

1. Computer Requirements

- 1.5 GHz processor or higher
- 2GB RAM or higher
- 500MB Available Hard Drive Space.
- Windows 7 SP2 or later operating system OR Any LINUX distribution.
- It is also Accessible on Mac OS
- An active Internet Connection(Not required for playing)

OR

2. Smartphone

- Android 9.0+(Web-view & WebGL Supported browser)

Technical Requirements

- C# (Primary Programming Language)
- .NET core 7 (Microsoft's .NET framework class library)
- Mono (Open-Sourced, .NET implementation)
- Unity3D (AAA, GameEngine)
- Unity Scripting API (Programming API to provide Access to core features)
- Git (Distributed version control system)
- GitHub (Git repository Hosting as well as Free basic website Hosting)

METHODOLOGY

While discussing the methodology to follow, among our team we were really confused until we looked back and thought what methodology does most successful game company uses.

Although we know this game is vastly different from a full-fledged cross-platform AAA games that are released by giants game companies, we realized the basic principle for making a game success is same for each of them.

We found that all of these company follow some a simple principle, that is making a game that they want to play, some companies completely focuses on it. So in the end we decided to take this tested path followed by layed and followed by many before us.

WHAT IS “Usability and user experience” Methodology?

Usability and user experience - Principle:

Usability means making products and systems easier to use, and matching them more closely to user needs and requirements. The

international standard, ISO 9241-11, provides guidance on usability and defines it as the extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use.

The phrase; “Usability and user experience” is Usability methods have been directly applied to games, but we need specific methods. For example, Nielsen’s usability heuristics have served as guidelines for creating usable applications. Each aspect of usability heuristics, as described by the ISO, is measured independently.

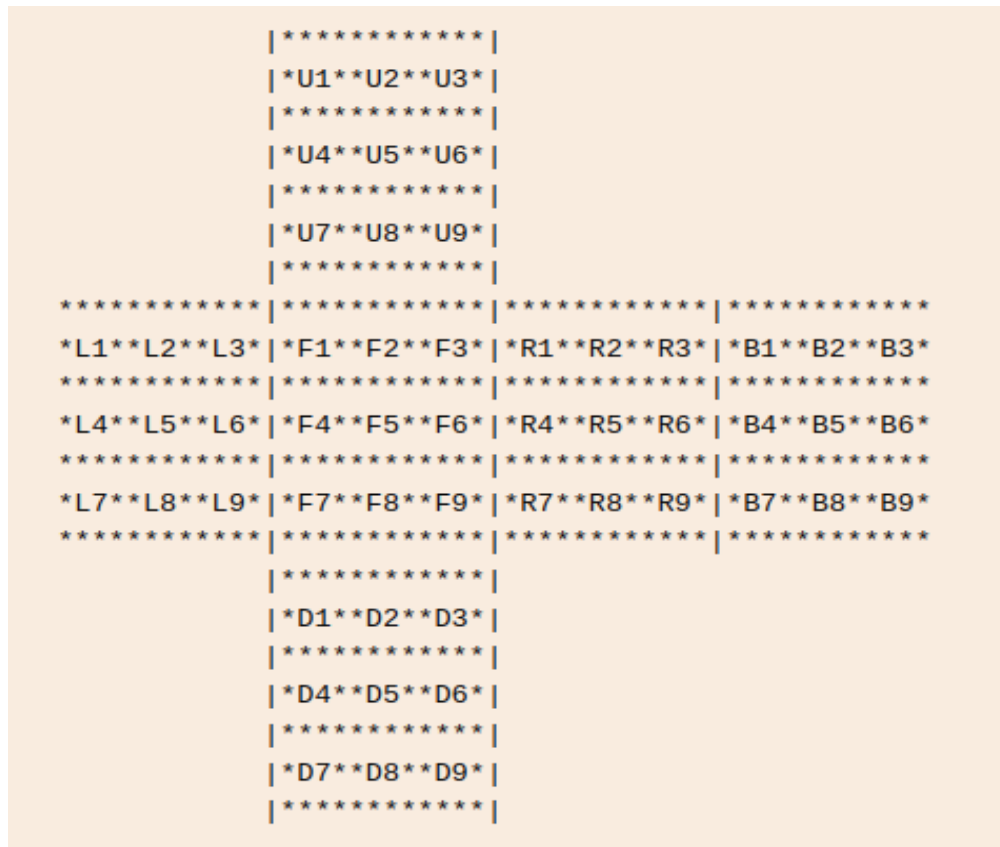
Game companies use mixed methodologies driven partly by their own internal needs and partly by the requirements of publishers. Every group has their own variants.

But in general they are often agile for a pre-production period and then effectively waterfall for production. This is assuming they have a hard ship date they are trying to hit.

by the requirements of publishers. Every group has their own variants.

Cube String notation Diagram

The names of the facelet positions of the cube (letters stand for Up, Left, Front, Right, Back, and Down):

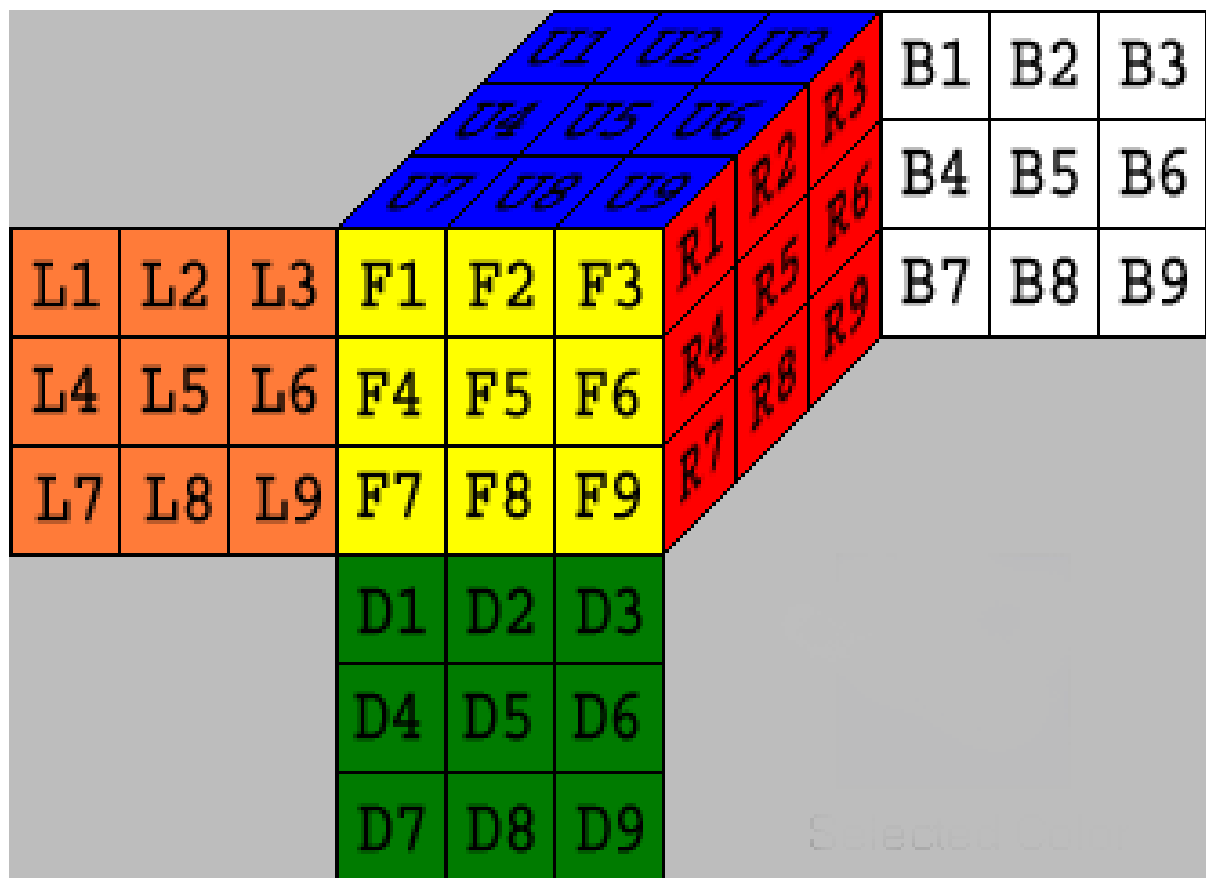


Solution string consists of space-separated parts, each of them represents a single move:

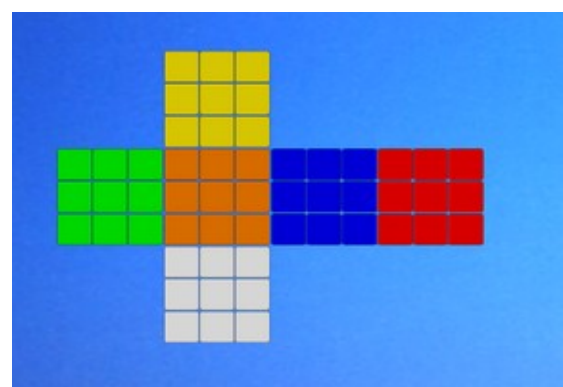
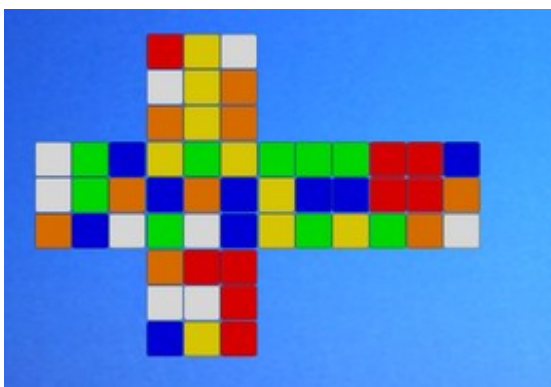
- A single letter by itself means to turn that face clockwise 90 degrees.
- A letter followed by an apostrophe means to turn that face counterclockwise 90 degrees.
- A letter with the number 2 after it means to turn that face 180 degrees.

e.g. R U R' U R U2 R' U

Cube Facelet Diagram



Based on above facelet diagram, i have created a cubemap so that user can have the view of complete cube at all times.



References

Technology References:

C# (<https://learn.microsoft.com/en-us/dotnet/csharp/>)

.NET Core (<https://dotnet.microsoft.com/en-us/learn/dotnet/what-is-dotnet>)

Mono(<https://www.mono-project.com/>)

Unity3D(<https://unity.com/>)

Unity Scripting API(<https://jquery.com/>)

LibreOffice(<https://www.libreoffice.org/discover/libreoffice/>)

Documentation References:

Unity Documentation (<https://docs.unity3d.com/Manual/index.html>)

kociemba Algorithm (<http://kociemba.org/>)

Others:

Git SCM(<https://git-scm.com/>)

GitHub(<https://github.com/>)

Composite reference:

Google (<https://www.google.co.in/>)

Project Repo(<https://github.com/Prakash4844/Rubikc>)

Team-members

Prakash (205028)

Sagar Goswami (205040)