

SPEED RACER
PROJECT REPORT

Submitted By

KARTHIKA SURESH (Reg NO: MCT20MCA-2023)

To

APJ Abdul Kalam Technological University

In partial fulfilment of the requirements for the award of the Degree of

MASTER OF COMPUTER APPLICATIONS



DEPARTMENT OF COMPUTER APPLICATIONS

MOHANDAS COLLEGE OF ENGINEERING & TECHNOLOGY

Anad, Nedumangad,

Thiruvananthapuram

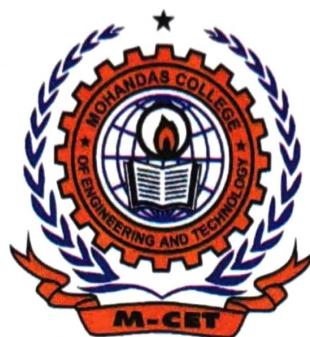
695544

2022

DEPARTMENT OF COMPUTER APPLICATIONS
MOHANDAS COLLEGE OF ENGINEERING &
TECHNOLOGY

Anad, Nedumangad, Thiruvananthapuram-695 544

CERTIFICATE



This is to certify that the report entitled "**SPEED RACER**" submitted by **Ms. KARTHIKA SURESH** (Register No: MCT20MCA-2023) to APJ Abdul Kalam Technological University in partial fulfilment of the requirements for the award of the degree **MASTER OF COMPUTER APPLICATIONS** is a bona fide record of the project work carried out by her under my guidance and supervision. This report in any form has not been submitted to any other University or Institute for any purpose.

Internal Supervisor(s)

Project Coordinator

8/9/22

Head of the Department

Asst. Professor, Dept. of MCA
Mohandas College of Engineering & Technology
Thiruvananthapuram

External Examiner

DECLARATION

I hereby declare that the project report “**SPEED RACER**”, submitted for partial fulfilment of the requirements for the award of degree of Master of Computer Application of the APJ Abdul Kalam Technological University, Kerala is a bona fide work done by me under supervision of prof Ms. JEEJA GS. I have adequately and accurately cited and referenced the original source. I also declare that I have adhered to ethics of academic honestly and integrity and have not misrepresented or fabricated any data or idea or fact or source in my submission. I understand that any violation of the above will be cause for disciplinary action by the institute and or the University and can also evoke penal action from the source which have thus not been properly cited or from whom proper permission has not been obtained. This report has not been previously formed the basis for the award of the degree, diploma or similar title of any other University

Place: Trivandrum

KARTHIKA SURESH

Date:

ACKNOWLEDGEMENT

At the outset, I thank God Almighty for standing by me throughout the project and making it possible for me to complete the project within the stipulated time.

I wish to record my deep sense of gratitude to our Director, **Dr. ASHALATHA THAMPURAN**, for her extensive support and guidance throughout the course of my project.

I wish to record my deep sense of gratitude to our Principal, **Dr. S. SHEELA**, for her extensive support and guidance throughout the course of my project.

I am thankful to our **HOD Ms. SREEJA K**, (Department of Computer Applications) for her kind co-operation and guidance throughout the course of my project.

I am also thankful to our Project Guide **Ms. JEEJA GS** (Department of Computer Applications) for him kind co-operation and guidance throughout the course of my project.

I also extend my sincere thanks to all other faculty members of Department of Computer Applications and our friends for their co-operation and encouragements.

BY,

KARTHIKA SURESH

INDEX

	TITLE	PAGE NO
1	ABSTRACT	
2	INTRODUCTION	1
3	SYSTEM SPECIFICATION	3
	3.1 Software Requirements	3
	3.2 Hardware Requirements	3
	3.3 Language Description	4
4	SYSTEM ANALYSIS	12
	4.1 Introduction	12
	4.2 Identification of Need	13
	4.3 Existing System	14
	4.4 Proposed System	17
	4.5 Proposed Methodology	18
5	FEASIBILITY STUDY	23
	5.1 Economic Feasibility	23
	5.2 Behavioural Feasibility	24
	5.3 Technical Feasibility	24
6	SYSTEM DESIGN	25
	6.1 Data Flow Diagram	25

7	TESTING	27
	7.1 Unit Testing	27
	7.2 Integration Testing	28
	7.3 Acceptance Testing	28
8	SYSTEM IMPLEMENTATION	29
9	SYSTEM MAINTENANCE	31
10	FUTURE ENHANCEMENTS	33
11	PRODUCT BACKLOG	34
12	SPRINT BACKLOG	36
13	SCREENSHOTS	37
14	CONCLUSION	42
15	REFERENCES	43

ABSTRACT

Game designing is a really interactive and creative part of modern IT culture. This has motivated various developers to create interesting games. The focus is on developing a 3D racing car game, using a process based upon agile development, an evolutionary development method.

It will be a single player game. After selecting the “Play game” option, the user can select the car type, for which there will be 3 options. The user can then select any of the following options, “Race Mode”, “Score Attack”, “Time Attack”. In Race Mode, the player will have to defeat the AI opponent in order to secure the first place. In Score Attack mode, the player should collect as many objects as possible in order to increase the score. In Time Attack mode, the best time taken to complete 3 laps should be displayed.

The platform of the game is Unity, which is a cross-platform game engine developed by Unity Technologies and main coding is in C# which is a general object-oriented programming (OOP) language for networking and Web development. Debugging is done in Visual Studio.

The whole game development process will cover implementation of real-time graphics, physics engine, sound effects and background music. Thus, the aim is to create a game which is extremely entertaining with great graphics to make it visually appealing.

INTRODUCTION

Video game development is a diverse field, it is a hybrid of game production and game design and has requisite skills from both fields forming the core of a video game developer's knowledge. A video game developer is usually a big picture position in the creation of a video game, guiding the project through multiple phases. That being the case, a video game developer is a cross between a producer and a programmer, they are a coordinating administrator with an artistic vision who also possesses the technical skill to oversee and contribute to software engineering, image rendering, editing and other aspects of game design.

Video games have been around for decades, providing entertainment for children and adults alike. They have evolved significantly from the early days of computer games and the first versions of Nintendo and Atari. The days of pixelated screens and limited sounds are a distant memory as video games have become more lifelike than ever. As technology continues to improve, so do video games. Non-traditional gaming companies, like Meta, Apple, and Google, are getting into the industry. The tech industry is looking for ways to make video game streaming as natural as streaming a song on Spotify or a movie on Netflix.

Microsoft (MSFT) has already been in the gaming industry through its popular Xbox console. In 2019, the company introduced Project xCloud, a video game streaming service that allows users to stream Microsoft's Xbox games to PCs or other devices. The service became fully functional in September 2020 and can be accessed by subscribers to Xbox Game Pass Ultimate.

The first video games were non-commercial, and were developed in the 1960s. They required mainframe computers to run and were not available to the general public. Commercial game development began in the 1970s with the advent of first-generation video game consoles and early home computers like the Apple I. Due to low costs and low capabilities of computers, a lone programmer could develop a full game. However, approaching the 21st century, ever-increasing computer processing power and heightened consumer expectations made it difficult for a single person to produce a mainstream console or PC game.

Mainstream PC and console games are generally developed in phases. First, in preproduction, pitches, prototypes, and game design documents are written. If the idea is approved and the developer receives funding, a full-scale development begins. This usually involves a team of 20–100 individuals with various responsibilities, including designers, artists, programmers, and testers. In 2016, Meta began developing a gaming platform with Unity Technologies, which makes a game development framework for people to make video games. Google launched its program, Stadia, which is Google's cloud gaming service, that allows users to play streaming video games with extremely high quality. It is available through Google's browser, phones, tablets, and desktop computers. The key goal for all of these companies is to allow players to stream video games without the need for a computer or a video game console. As this trend continues, purchasing physical video games in the form of cartridges or discs is becoming increasingly rare.

The video game industry has always been about innovation. New technology, new controls, and new experiences are to be expected. As the world moves more and more to time spent on mobile phones, streaming services, and mobile phone game playing will become an important arena for revenues, and large tech companies will look to leverage their current framework to get involved.

SYSTEM SPECIFICATION

3.1 SOFTWARE REQUIREMENTS

- Windows 7 or higher
- DirectX 11 or above

3.2 HARDWARE REQUIREMENTS

- CPU : Intel i5 3rd Generation/AMD Ryzen 3 or above
- RAM :4GB
- GPU : Intel HD 4000/Nvidia GT 710/AMD HD 3450 or above
- Disk Space: At least 5 GB

3.3 LANGUAGE DESCRIPTION

3.3.1 C#

C# is a general object-oriented programming (OOP) language for networking and web development. It is specified as a common language infrastructure (CLI) language.

In January 1999, Dutch software engineer Anders Hejlsberg formed a team to develop C# as a complement to Microsoft's .NET framework. Initially, C# was developed as CLike Object Oriented Language (Cool). The actual name was changed to avert potential trademark issues. In January 2000, .NET was released as C#. Its .NET framework promotes multiple Web technologies.

C# is used for:

- Mobile applications
 - Desktop applications
 - Web applications
 - Web services
 - Web sites
 - Games
 - VR
 - Database applications
- Advantages of using C#:
- It is one of the most popular programming languages in the world
 - It is easy to learn and simple to use
 - It has a huge community support
 - C# is an object-oriented language which gives a clear structure to programs and allows code to be reused, lowering development costs
 - As C# is close to C, C++ and Java, it makes it easy for programmers to switch to C# or vice versa

The C# programming language was designed by Anders Hejlsberg from Microsoft in 2000 and was later approved as an international standard by Ecma (ECMA-334) in 2002 and ISO/IEC (ISO/IEC 23270) in 2003. Microsoft introduced C# along with .NET Framework and Visual Studio, both of which were closed-source.

Since the release of C# 2.0 in November 2005, the C# and Java languages have evolved on increasingly divergent trajectories, becoming two quite different languages. One of the first major departures came with the addition of generics to both languages, with vastly different implementations. C# makes use of reification to provide "first-class" generic objects that can be used like any other class, with code generation performed at class-load time. Furthermore, C# has added several major features to accommodate functional-style programming, culminating in the LINQ extensions released with C# 3.0 and its supporting framework of lambda expressions, extension methods, and anonymous types. These features enable C# programmers to use functional programming techniques, such as closures, when it is advantageous to their application. The LINQ extensions and the functional imports help developers reduce the amount of boilerplate code that is included in common tasks like querying a database, parsing an xml file, or searching through a data structure, shifting the emphasis onto the actual program logic to help improve readability and maintainability. C# used to have a mascot called Andy (named after Anders Hejlsberg). It was retired on January 29, 2004.

C# was originally submitted to the ISO/IEC JTC 1 subcommittee SC 22 for review, under ISO/IEC 23270:2003, was withdrawn and was then approved under ISO/IEC 23270:2006. The 23270:2006 is withdrawn under 23270:2018 and approved with this version.

3.3.2 JavaScript

JavaScript often abbreviated as JS, is a high-level, interpreted programming language. It is a language which is also characterized as dynamic, weakly typed, prototype-based and multi-paradigm. JavaScript is a lightweight, interpreted programming language. It is designed for creating network-centric applications. It is complimentary to and integrated with Java. JavaScript is very easy to implement because it is integrated with HTML. It is open and cross-platform. Alongside HTML and CSS, JavaScript is one of the three core technologies of the World Wide Web. JavaScript enables interactive web pages and thus is an essential part of web

applications. The vast majority of websites use it, and all major web browsers have a dedicated JavaScript engine to execute it.

JavaScript is a high-level, often just-in-time compiled language that conforms to the ECMAScript standard. It has dynamic typing, prototype-based object-orientation, and first-class functions. It is multi-paradigm, supporting event-driven, functional, and imperative programming styles. It has application programming interfaces (APIs) for working with text, dates, regular expressions, standard data structures, and the Document Object Model (DOM).

Although, JavaScript has no connectivity with Java programming language. The name was suggested and provided in the times when Java was gaining popularity in the market. In addition to web browsers, databases such as CouchDB and MongoDB uses JavaScript as their scripting and query language. Unity compiles your sources into JavaScript from C/C++ code using emscripten, so you can also write plugins in C/C++ code, and call these functions from C#.

JavaScript in Unity

- Helps to write logic of the game in a multi-paradigm, dynamically typed language with strong concepts of meta-programming, where beautiful architecture can be created, and creativity can be unleashed when coding the game world without losing focus on technical stuff.
- Game scripts logic can be abstracted from lower-level engine logic, also allowing to write automated tests for your story, dialogs and interactions, without even running Unity engine.
- Easily expose your game logic to the community, so fans can create mods and addons.
- Make your game portable to any other engine than Unity, if needed
- Have access to the npm library with thousands of free JavaScript libraries and tools

Features of JavaScript

1. All popular web browsers support JavaScript as they provide built-in execution environments.
2. JavaScript follows the syntax and structure of the C programming language. Thus, it is a structured programming language.
3. JavaScript is a weakly typed language, where certain types are implicitly cast (depending on the operation).
4. JavaScript is an object-oriented programming language that uses prototypes rather than using classes for inheritance.
5. It is a light-weighted and interpreted language.
6. It is a case-sensitive language.
7. JavaScript is supportable in several operating systems including, Windows, macOS, etc.
8. It provides good control to the users over the web browsers.

3.3.3 Game Engine- Unity 3D

To save time in our development process, we decided that Unity would be the most suited game engine for our needs.

Unity is a cross platform game engine developed by Unity Technologies and is used to develop video games for PC, consoles, mobile devices and websites. As well as a game engine, Unity is an IDE. IDE stands for “integrated development environment,” which describes an interface that gives you access to all the tools you need for development in one place. The Unity software has a visual editor that allows creators to simply drag and drop elements into scenes and then manipulate their properties. Unity gives users the ability to create games in both 2D and 3D, and the engine offers a primary scripting API in C#, for both the Unity editor in the form of plugins, and games themselves, as well as drag and drop functionality. Debugging is done in Visual Studio. Visual Studio brings a premium debugging experience to the Unity

game engine. Identify issues quickly by debugging your Unity games in Visual Studio— Set breakpoints and evaluate variables and complex expressions. C# is also one of the more beginner-friendly programming languages. And it's well worth learning, as it is widely used in the industry and also shares a lot in common with other popular languages such as C and Java.

Within 2D games, Unity allows importation of sprites and an advanced 2D world renderer. For 3D games, Unity allows specification of texture compression and resolution settings for each platform that the game engine supports and provides support for bump mapping, reflection mapping, parallax mapping, screen space ambient occlusion (SSAO), dynamic shadows using shadow maps, render-to-texture and full-screen post-processing effects. Unity also offers services to developers, these are: Unity Ads, Unity Analytics, Unity Certification, Unity Cloud Build, Unity Every play, Unity IAP, Unity Multiplayer, Unity Performance Reporting, Unity Collaborate and Unity Hub.

Unity 3D has a free as well as a pro version with several features attached to it. Initially, choose the version which is free and get the major gaming features highlights. Further, the user can utilize the distinct focal points with the Pro-Version for developing games with high-end features like sound channel, 3D composition booster, feature playback etc. Unity 3D game development gives permission to all sorts of developers to make fullfledged full version games without having to rev up costs of any kind. It is very easy to create several games for all the platforms and design several stages of the games without any hassle. Porting to the next stage is very easy with Unity compared to other platforms. Each stage of the game has got some interesting elements and with Unity you can transport those elements to the next stage and make it more interesting than before. The unity asset store is a one-stop destination for all the developers to fulfil their gaming requirements. It provides developers with all the requirements for creating your game effortlessly without letting you stuck while creating characters, backdrops, buildings, and so on. Also, with the asset store there is an opportunity for musical artists, craftsmen, or modelers to earn more money. Hence, the store also creates multiple money-making opportunities. Although, Unity 3D is known for 3D games, however, the platform is also very effective in creating engaging 2D Games for mobiles, PC's and even for gaming consoles. Unity 4.3 has an implicit 2D motor which helps game developers

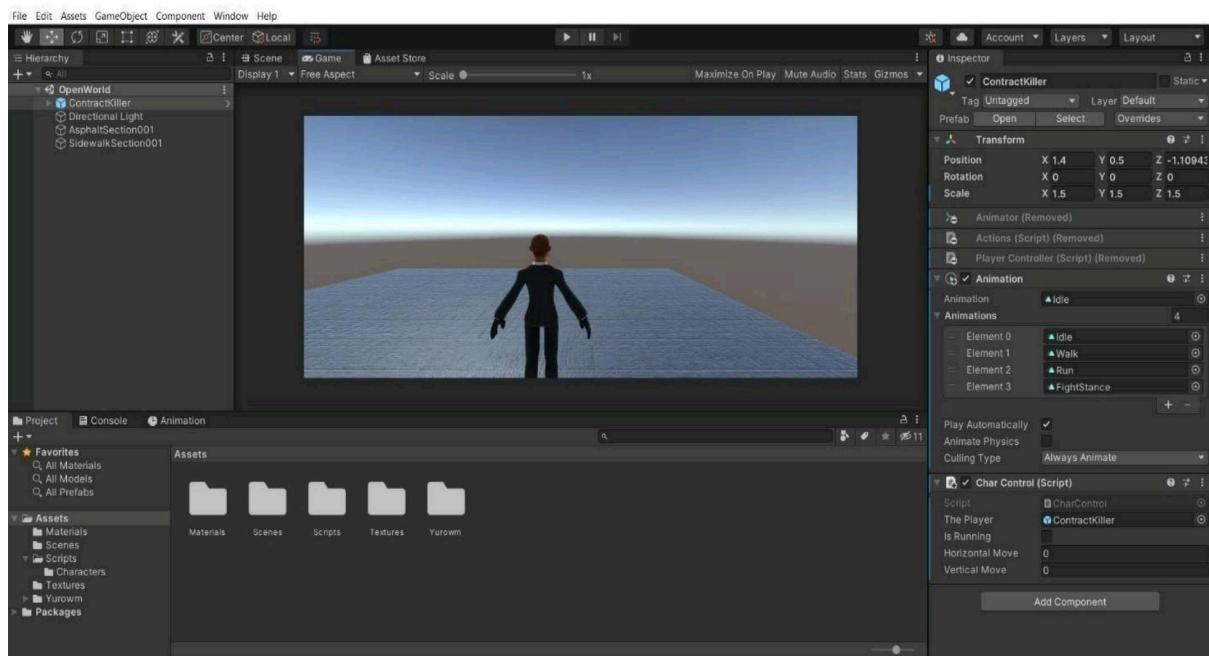
in developing effective and efficient 2D Games. The developer can easily integrate the physical science of a 2D world and create mind-blowing games with eye-catching features. With Unity game engine, it is extremely easy to create multiplayer games. This is because the game engine offers multiple features that motivates developers to develop mind-blowing games for players all across the internet. One of the best examples is Solstice Arena which is very popular multiplayer game that has a huge fan base all across the web. The whole game development process is very challenging but with Unity the process becomes easy, and fun.

Advantages of Unity

- ◆ Unity has a broad reach that makes recruitment easier
- ◆ Unity is a good engine for beginners
- ◆ Unity is fast and agile
- ◆ Unity makes portability easier
- ◆ Unity has a large and varied asset store
- ◆ Unity allows you to build your own tools
- ◆ Unity is good for VR developers



Tea for Two's Snowtopia





RageSquid and No More Robots' Descenders

SYSTEM ANALYSIS

4.1 INTRODUCTION

System/Requirement analysis phase is considered to be one of the most important phases in the system development life cycle. It is immensely important that the software developer make a thorough study of the existing systems. It refers into the process of examining situations with the intent of improving it through better procedures and methods. System Analysis is the process of planning a new system to either replace or complement an existing system. But before any planning is done, the old system must be thoroughly understood and the requirements must be understood. System Analysis is therefore, the process of gathering and interpreting facts, diagnosing problems and using the information to make improvements in the system. System analysis is done with the following objectives in mind:

- Identification of public need
- Evaluation of the system concept for feasibility - Economic and Technical analysis

4.2 IDENTIFICATION OF NEED

In today's market of games, we noticed a distinct lack of local co op games, even though this feature was cherished by many while we were young. This is why we intended to make a fun arcade car racing game that will keep you at the edge of your seat.

An untapped market in the category of arcade gaming was recognized and chose to act on it. Games nowadays rely too much on microtransactions and fails to focus on the gameplay.

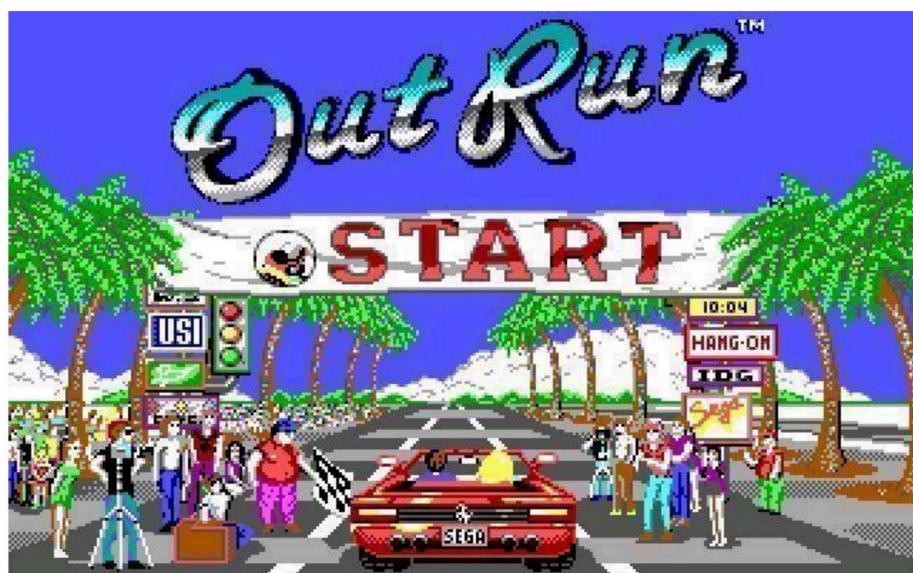
4.3 EXISTING SYSTEM

A number of Arcade Racing games are already present in the market. However most of them are way past their lifecycle and there is a distinct lack of games in recent years in the Arcade Racing games market of the videogame industry. A distinct feature of arcade racing has been its lenient physics allowing for a more “fun” approach to racing, where players don’t have to brake as often and can even corner at breakneck speeds without much consequence. Games like Outrun by SEGA and Midnight Club by Rockstar games are considered classics in the genre. They feature thrilling racing action but are sorely lacking in visual quality by today’s standards.

4.3.1 Outrun by SEGA

Outrun is an Arcade Racing game that was released in 1986 by SEGA. The game was a critical and commercial success, becoming one of the bestselling titles of its time. In the game, the player controls a car from the third-person perspective and is required to reach the destination before a timer runs out. Outrun is one of the most influential racing games and have spawned many sequels, but none have captured the fun of the original game. While Outrun’s graphics were legendary for its time, it’s quite weak when compared to modern racing titles.





4.3.2 Need For Speed II by EA games

The Need for Speed games are one of the most well-known racing game franchises in the industry. However, in the last 2 decades, they have been focusing on realism and plot driven games and as such have not put out any Arcade style game in a long time. NFS II was released in 1997 and is the second instalment in the NFS franchise. Players can race against computer-controlled opponents or against other players in Split-Screen or LAN play.



4.4 PROPOSED SYSTEM

Aim is to make a classic arcade racing game with photorealistic graphics. Unity UX is used to build a clean, simple and balanced UX to create an game application interface. The proposed game is single player. After selecting the “Play game” option, the user can select the car type, for which there will be 3 options. The user can then select any of the following options, “Race Mode”, “Score Attack”, “Time Attack”. In Race Mode, the player will have to defeat the AI opponent in order to secure the first place. In Score Attack mode, the player should collect as many objects as possible in order to increase the score. In Time Attack mode, the best time taken to complete 3 laps should be displayed.

4.5 PROPOSED METHODOLOGY

Agile method proposes incremental and iterative approach to software design. The agile process is broken into individual models that designers work. The customer has early and frequent opportunities to look at the product and make decision and changes to the project.

- Agile model is considered unstructured compared to the waterfall model.
- Small projects can be implemented very quickly. For large projects, it is difficult to estimate the development time.
- Error can be fixed in the middle of the project.
- Documentation attends less priority than software development.
- Every, iteration has its own testing phase. It allows implementing regression testing every time new functions or logic are released
- In agile testing when an iteration end, shippable features of the product is delivered to the customer. New features are usable right after shipment. It is useful when you have good contact with customers.
- Testers and developers work together.
- At the end of every sprint, user acceptance is performed
- It requires close communication with developers and together analyses requirements and planning.

Agile is not only about applying the set practices in developing a software. It also brings in a change in the team's mind-set, which drives them towards building a better software, working together and eventually landing them a happy customer. Agile values and principles enable the team to shift their focus and change their thought process of building a better software.

SCRUM

SCRUM is an agile development method which concentrates specifically on how to manage tasks within a team-based development environment. Basically, scrum is derived from activity that occurs during a rugby match. Scrum believes in empowering the development team and advocates working in small teams. The outcome is a version of scrum that is unique and specific, in order to have a process that works for us. Scrum is part of the agile movement. Agile is a response to the failure of the dominant software development project management paradigms and borrows many principles from lean manufacturing. The agile manifesto placed a new emphasis on communication and collaboration, functioning software, team selforganization, and the flexibility to adapt to emerging business realities.

Scrum's early advocates were inspired by empirical inspect and adapt feedback loops to cope with complexity and risk. Scrum emphasizes decision making from real world results rather than speculation. Time is divided into short work cadences, known as sprints, typically one week or two weeks long. The product is kept in a potentially shippable state at all times. Scrum is a simple set of roles, responsibilities and meetings that never change. It consists of three roles, and their responsibilities are explained as follows:

SCRUM MASTER

- This role is played by HOD, Ms. Sreeja K
- Master is responsible for setting up the team, sprint meeting and removes obstacles to progress.
- Helps everyone involved understand and embrace the scrum values, principles, and practices.
- As a facilitator, scrum master helps the team resolve issues and make improvements Its use of scrum.
- She is responsible for protecting the team from outside interference.
- She takes a leadership role in removing impediments that inhibit team Productivity.
- Acts as a coach, providing development process leadership.

PRODUCT OWNER

- Role played by Ms. Jeeja GS (Project Guide)
- The product owner creates product backlog, prioritizes the backlog and is responsible for the delivery of the functionality at each iteration.
- She maintains and communicates to all other participants a clear vision of what the scrum team is trying to achieve.
- She is the only authority responsible for what will be developed and in what order.
- One of the most important responsibilities of product owner is to manage product backlog.

SCRUM TEAM

Team manages its own work and organizes the work to complete the sprint or cycle. The team is a self- organizing and cross-functional group of people who do the hands-on work of developing and testing the product. Since the team is responsible for producing the product, it must also have the authority to make decisions about how to perform the work. The team is therefore self- organizing: team members decide how to break work into tasks, and how to allocate tasks to individuals, throughout the Sprint. The team size should be kept in the range from five to nine people, if possible (a large number make communication difficult, while a smaller number leads to low productivity and fragility) Note:

A very similar term, “scrum Team,” refers to the team plus the scrum master and product owner.

SPRINT

The sprint backlog is a list of tasks identified by the scrum team to be completed during the scrum sprint. During the sprint planning meeting, the team selects some number of products backlog items, usually in the form of user stories, and identifiers the tasks necessary to complete each user stories. Most teams also estimate how many hours each task will take someone on the team to complete. It's critical that the team selects the items and size of the sprint backlog. Because they are the people committing to completing the tasks, they must be the people to choose what they are committing to during the scrum sprint. The sprint backlog is commonly maintained as a spreadsheet, but it is also possible to use your defect tracking system or any of a number of software products designed specifically for scrum or agile. The sprint backlog makes visible all the work that the Development Team identifies as necessary to meet the sprint goal. To ensure continuous improvement, includes at least one high priority process improvement identified in the previous Retrospective meeting. The sprint backlog is a plan with enough detail that changes in progress can be understood in the daily scrum. The Development Team modifies the sprint backlog throughout the sprint. Only the Development Team can change its sprint backlog during a sprint. The sprint backlog is a highly visible, real time picture of the work that the Development Team.

DEVELOPMENT PROCESS

- Pre-production
- High concept
- Pitch
- Game design document and presentation
- Production design
- Programming
- AI Pathfinding
- Audio and art production
- Testing
- First playable
- Release
- Maintenance

FEASIBILITY STUDY

Feasibility study is a set of system proposal according to its workability, impact on the organization, ability to meet user needs and effective use of resources. The objective of feasibility study is not to solve the problem, but to acquire a sense of its scope. Three key considerations are involved in the feasibility analysis. These are:

- Economic Feasibility
- Behavioural Feasibility
- Technical Feasibility
-

5.1 Economic Feasibility

Economic analysis is the most frequently used method for evaluating the effectiveness of the candidate system. This is more commonly known as cost/benefit analysis. The procedure is to determine the benefits that are obtained from the candidate system and compare that to the cost of building the system

Software product development is a paradigm shift from routine application maintenance and support in the software industry. Development a game/software product from scratch is a significant challenge for any organization. It requires considerable investments in terms of effort and cost and also confirms client involvement, knowledge about client markets like Steam, Google Play, etc.

We target a demographic of people left behind in the modern racing game market - The Arcade Racers. Arcade-style racing games put fun and a fast-paced experience above all else, as cars usually compete in unique ways. A key feature of arcade-style racers that specifically distinguishes them from simulation racers is their far more liberal physics. Arcade racing games used to be popular in the late 90s and early 00s but have been left behind in the modern game market with Video Game companies pushing for more cinematic games over straightforward arcade racing action. We are adding our own touch to the genre with a progression system that allows the player to unlock more cars as you play more.

The video game industry is growing at an incredible pace, projected to be worth an astronomical 120 billion USD by 2019. As such, we can get at least a guaranteed return on investment with this kind of growing market.

5.2 Behavioural Feasibility

It includes how strong the reaction of staff will be towards the development of new system that involves computer's use in their daily work. In this case, we see how the player reacts to a new game in the genre.

5.3 Technical Feasibility

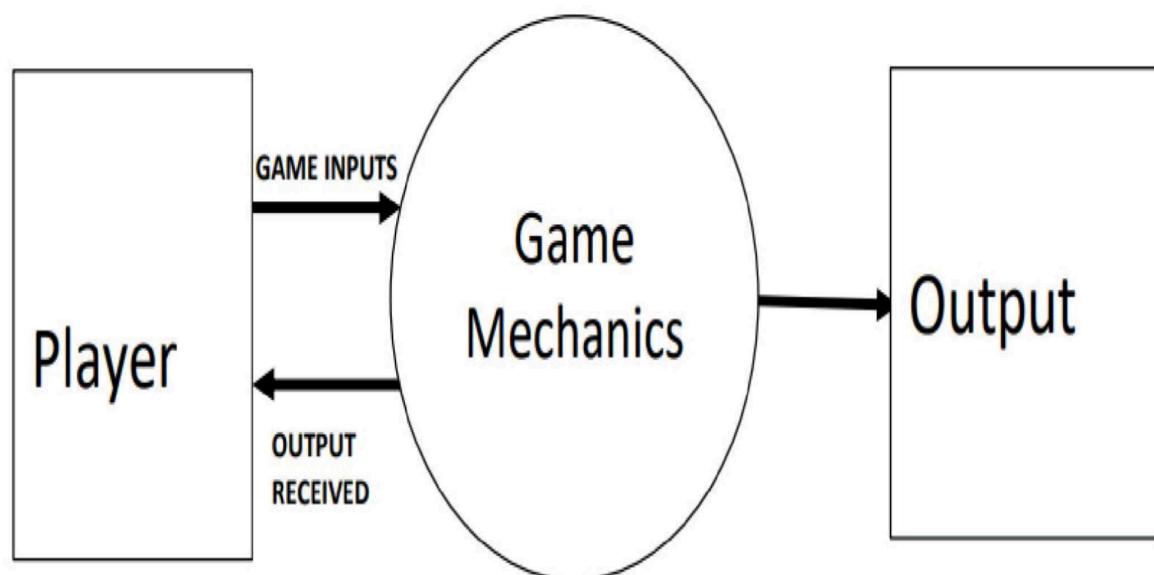
It checks whether the existing computer systems support the candidate system or not or up to what extent it supports. In this case, we have fairly low system requirements in order to run the game at an acceptable frame rate. This is done to support a large number of systems and hence helps in gathering a wide audience.

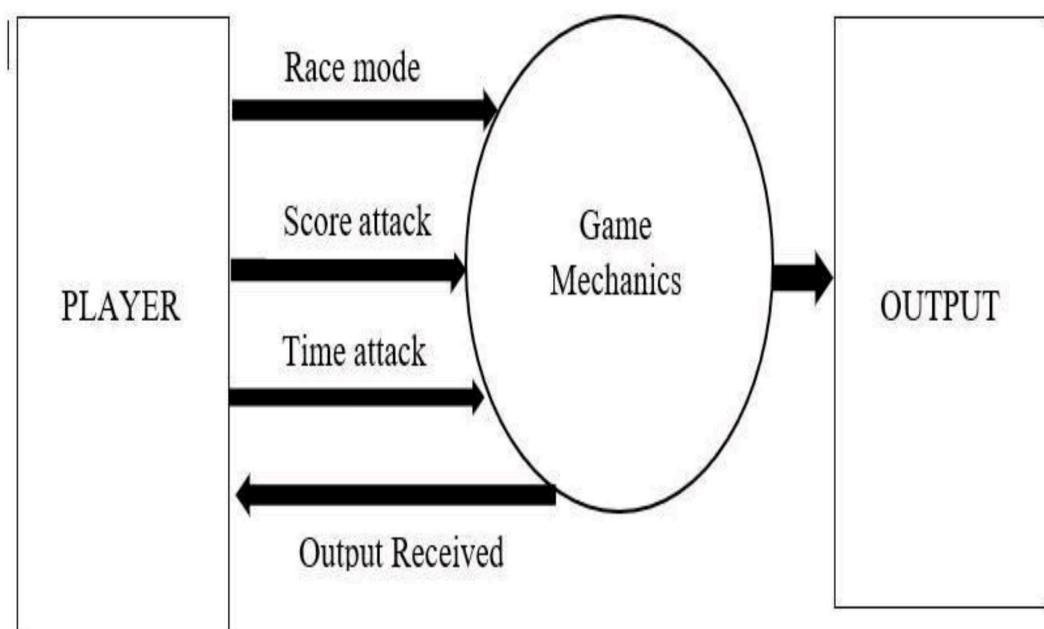
However, if the user is in possession of a high-end system, he will be able to run the game with a very high visual quality and high frame rates, making for a more visually appealing game.

SYSTEM DESIGN

6.1 Data Flow Diagram

LEVEL 0



LEVEL 1

TESTING

7.1 Unit Testing

It is a level of software testing where individual units/ components of software are tested. The purpose is to validate that each unit of the software performs as designed. A unit is the smallest testable part of any software. It usually has one or a few inputs and usually a single output. In procedural programming, a unit may be an individual program, function, procedure, etc. In object-oriented programming, the smallest unit is a method, which may belong to a base/ super class, abstract class or derived/ child class. (Some treat a module of an application as a unit. This is to be discouraged as there will probably be many individual units within that module.) Unit testing frameworks, drivers, stubs, and mock/ fake objects are used to assist in unit testing.

Unit testing increases confidence in changing/ maintaining code. If good unit tests are written and if they are run every time any code is changed, we will be able to promptly catch any defects introduced due to the change. Also, if codes are already made less interdependent to make unit testing possible, the unintended impact of changes to any code is less.

Codes are more reusable. In order to make unit testing possible, codes need to be modular. This means that codes are easier to reuse. Writing tests takes time but the time is compensated by the less amount of time it takes to run the tests; You need not fire up the GUI and provide all those inputs. The effort required to find and fix defects found during unit testing is very less in comparison to the effort required to fix defects found during system testing or acceptance testing. The cost of fixing a defect detected during unit testing is lesser in comparison to that of defects detected at higher levels. Debugging is easy. When a test fails, only the latest changes need to be debugged.

7.2 Integration Testing

Integration testing is the phase in software testing in which individual software modules are combined and tested as a group. Integration testing is conducted to evaluate the compliance of a system or component with specified functional requirements. It occurs after unit testing and before validation testing. Integration testing takes as its input modules that have been unit tested, groups them in larger aggregates, applies tests defined in an integration test plan to those aggregates, and delivers as its output the integrated system ready for system testing.

7.3 Acceptance Testing

Acceptance testing is also known as user acceptance testing (UAT), end-user testing, Operational Acceptance Testing (OAT) or field (acceptance) testing. Acceptance criteria are the criteria that a system or component must satisfy in order to be accepted by a user, customer, or other authorized entity. User acceptance testing consists of a process of verifying that a solution works for the user. It is not system testing (ensuring software does not crash and meets documented requirements), but rather ensures that the solution will work for the user (i.e., tests that the user accepts the solution); software vendors often refer to this as "Beta testing".

SYSTEM IMPLEMENTATION

System implementation uses the structure created during architectural design and the results of system analysis to construct system elements that meet the stakeholder's requirements and system requirements developed in the early lifecycle phases. These system elements are then integrated to form intermediate aggregates and finally the complete system-of-interest (SOI).

Implementation is the process that actually yields the last level system elements in the system hierarchy (System Breakdown Structure). System elements are made reused. Production involves the hardware fabrication process of forming, removing, joining, and finishing, the software realization process of coding and testing, or the operational procedures development process for operator's roles. If implementation involves a production process, a manufacturing system which uses the established technical and management process may be required.

The purpose of the implementation process is to design and create (or fabricate) a system element confirming to those elements design properties and requirements. The element is constructed employing appropriate technologies and industry practices. This process bridges the system definition process and the integration process.

The following major activities and tasks are performed during this process:

- Define the implementation strategy - Implementation process activities begin with detailed design and include developing an implementation strategy that defines fabrication and coding procedures, tools and equipment to be used, implementation tolerances, and the means and criteria for auditing configuration of resulting elements to the detailed design documentation. In the case of repeated system element implementations (such as for mass manufacturing or replacement elements), the implementation strategy is defined and refined to achieve consistent and repeatable element production; it is retained in the project decision database for future use. The implementation strategy contains the arrangements for packing, storing, and supplying the implemented element.

- Realize the system element - Realize or adapt and produce the concerned system element using the implementation strategy items as defined above. Realization or adaptation is conducted with regard to standards that govern applicable safety, security, privacy, and environmental guidelines or legislation and the practices of the relevant implementation technology. This requires the fabrication of hardware elements, development of software elements, definition of training capabilities, drafting of training documentation, and the training of initial operators and maintainers.
- Provide evidence of compliance - Record evidence that the system element meets its requirements and the associated verification and validation criteria as well as the legislation policy. This requires the conduction of peer reviews and unit testing, as well as inspection of operation and maintenance manuals. Acquire measured properties that characterize the implemented element (weight, capacities, effectiveness, level of performance, reliability, availability, etc.).

SYSTEM MAINTENANCE

The results obtained from the evaluation process help the organization to determine whether its information systems are effective and efficient or otherwise. The process of monitoring, evaluating, and modifying of existing information systems to make required or desirable improvements may be termed as System Maintenance.

System maintenance is an ongoing activity, which covers a wide variety of activities, including removing program and design errors, updating documentation and test data and updating user support. For the purpose of convenience, maintenance may be categorized into three classes, namely:

i) Corrective Maintenance

Corrective Maintenance is a maintenance activity undertaken to overcome the failure or damage found during the preventive maintenance period. In general, corrective maintenance is not a scheduled maintenance activity, because it is done after a component is damaged and aims to restore the reliability of a component or system to its original state. Corrective maintenance also is known as breakdown or run to failure maintenance. Maintenance is only done after equipment or machine is damaged. If this maintenance strategy is used as the main strategy will result in a high impact of unplanned maintenance activities and replacement parts inventory.

ii) Adaptive Maintenance

Adaptive maintenance is the implementation of changes in a part of the system, which has been affected by a change that occurred in some other part of the system. Modification of a software product performed after delivery to keep a software product usable in a changed or changing environment. Adaptive maintenance consists of adapting software to changes in the environment such as the hardware or the operating system. The term environment in this context refers to the conditions and the influences which act (from outside) on the system. For example, business rules, work patterns, and government policies have a significant impact on the software system.

iii) Perfective Maintenance

Perfective maintenance means adding new programs or modifying the existing programs to enhance the performance of the information system. This type of maintenance is undertaken to respond to user's additional needs which may be due to the changes within or outside of the organization. Perfective maintenance mainly deals with implementing new or changed user requirements. Perfective maintenance involves making functional enhancements to the system in addition to the activities to increase the system's performance even when the changes have not been suggested by faults. This includes enhancing both the function and efficiency of the code and changing the functionalities of the system as per the users' changing needs.

iv) Preventive Maintenance

Preventive maintenance is performed on a schedule, which detects and lessens the degradation of components and systems. Preventative maintenance has the aim of increasing reliability, saving costs from major failure or down time, reducing the risk of failure and extending the life of machine or facilities. Preventive maintenance cannot assure escape from catastrophic failures and sometimes may seem to be unnecessary. Preventative maintenance includes test operations as per design, diagnosis, replacing worn out parts, measurements, lubricating, tightening and adjustments

FUTURE ENHANCEMENTS

Multi-player can easily be implemented into the system either locally through split screen or online. Player will be able to login to the system and race against each other. Future updates can bring more maps, cars and other game modes like sprint and circuit racing. Additional Features like an online leader board is also planned. Weather conditions that change and dynamically affect car control. As the system is highly modular, it is relatively easy to add and modify components as needed.

PRODUCT BACKLOG

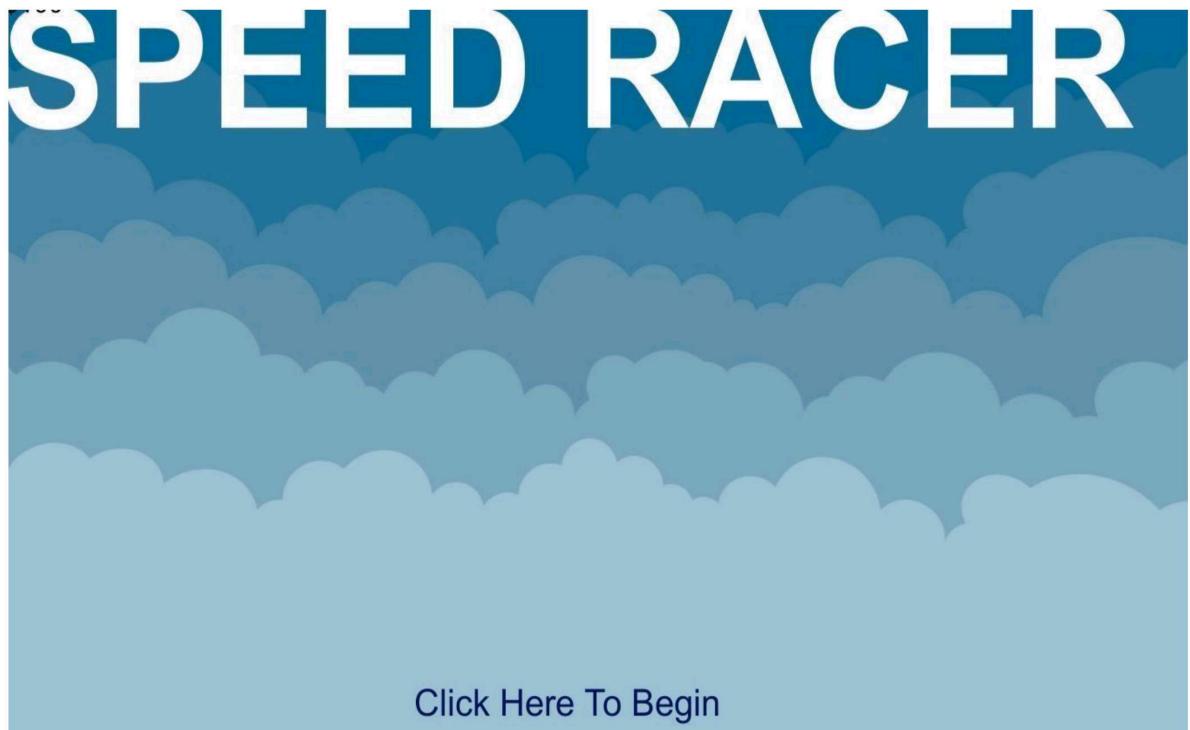
ID	USER STORIES	PRIORITY	COMMENT FROM SCRUM MASTER	COMMENT FROM PRODUCT OWNER
1	The camera should stay oriented when it collides with enemies and other obstructions	Very high		A third person perspective is provided while driving the car
2	The player should be able to control the car using arrow keys	Very high		Game controls shouldn't be too complicated
3	It should be possible to see graphics and art design that compiles with the game's universe	High		Very important
4	The user should be able to choose from 3 modes, which are, “Race mode”, “Time attack”, and “Score attack”	High		
5	In “Race mode”, there should be an enemy AI and it should not interfere with the player’s gaming experience	High		

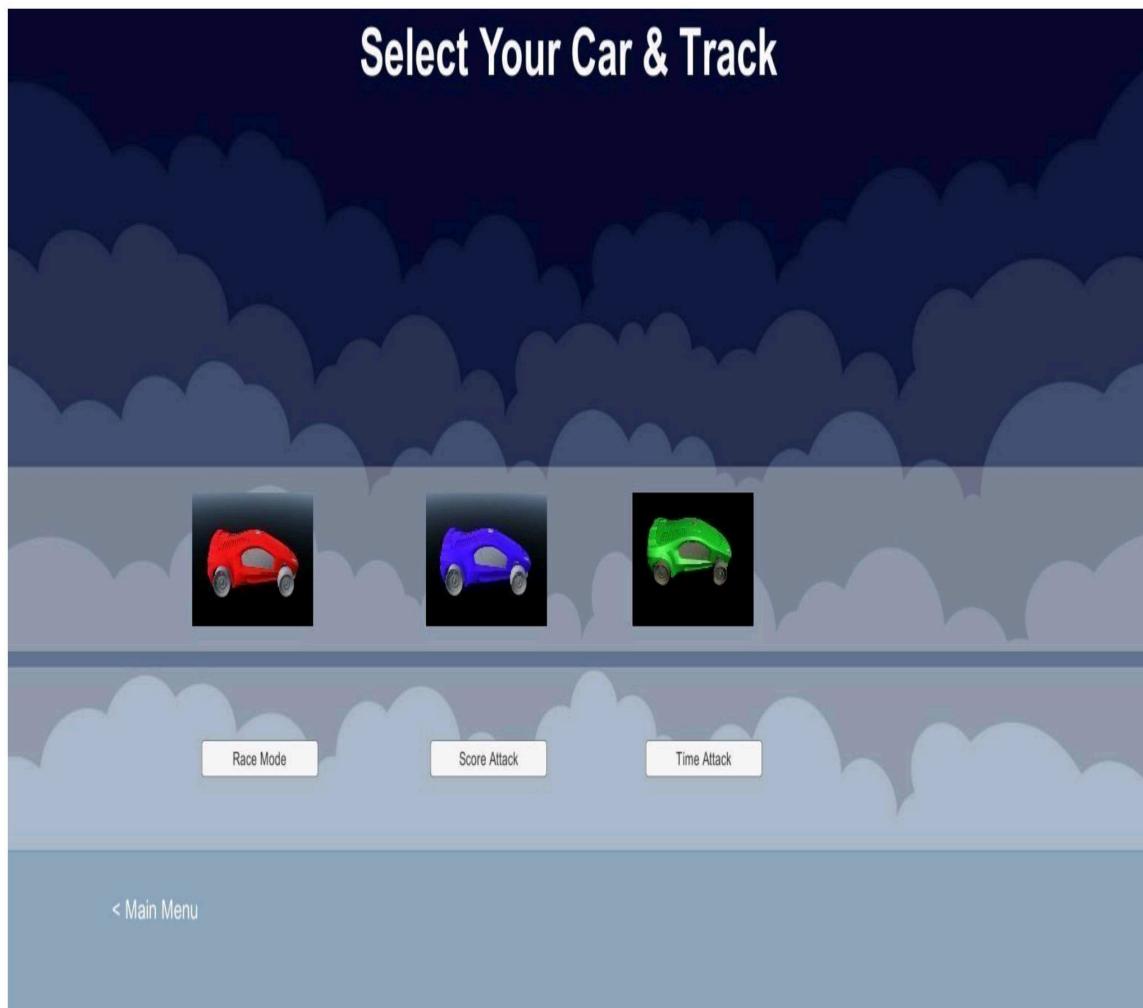
6	3 laps should be completed in order to win in Race mode.	High		
7	In order to increase the score, player should collect as many objects as possible in “Score attack” mode	High		
8	In “Time attack” mode, the best time taken to complete 3 laps should be displayed	High		
9	Documents should be done based on the study and outcomes generated	High		

SPRINT BACKLOG

USER STORIES	NOT STARTED	IN PROGRESS	COMPLETED
Creating game environment			Completed
Assets and skybox preparation			Completed
Background music			Completed
Programming/script			Completed
Adjust racing navigations and AI pathfinding for enemy AI			Completed
Documentation			Completed

SCREENSHOTS





RACE MODE



SCORE ATTACK



TIME ATTACK



CONCLUSION

At the end of the project, a videogame that is both graphically and mechanically appealing to a wide audience has been created. It can easily be ported to other systems in order to widen the market reach and also be put up for sale on videogame markets like Steam and the Google Play Store.

REFERENCES

1. https://www.youtube.com/channel/UCRMXHQ2rJ9_0CHS7mhL7erg/about
2. <https://www.youtube.com/watch?v=MQ5GJPIAGS4&list=PLZ1b66Z1KFKgkE9ji0tF2iDO0LGxmlwIm>
3. <https://unity.com/solutions/gaming-services>
4. <https://www.udemy.com/course/devslopes-unity3d/>
5. <https://unitycodemonkey.com/>
6. [https://docs.microsoft.com/en-us/archive/msdnhttps://docs.microsoft.com/enus/archive/msdnmagazine/2014/august/unityhttps://docs.microsoft.com/enus/archive/msdnhttps://s.microsoft.com/en-us/archive/msdnmagazine/2014/august/unity-developing-your-firstgame-with-unity-and-csharp-#:~:text=Unity%20is%20a%20native%20C%20Mono%20or%20the%20Microsoft%20magazine/2014/august/unity-developing-your-first-game-with-unity-and-csharp-#:~:text=Unity%20is%20a%20native%20C%20Mono%20or%20the%20Microsoft%20developing-your-first-game-with-unity-and-csharp#:~:text=Unity%20is%20a%20native%20C%20Mono%20or%20the%20Microsoft%20M%20icrosoft%20.](https://docs.microsoft.com/en-us/archive/msdnhttps://docs.microsoft.com/enus/archive/msdnmagazine/2014/august/unity-developing-your-first-game-with-unityand-csharp-#:~:text=Unity%20is%20a%20native%20C%20Mono%20or%20the%20Microsoft%20magazine/2014/august/unityhttps://docs.microsoft.com/enus/archive/msdnhttps://s.microsoft.com/en-us/archive/msdnmagazine/2014/august/unity-developing-your-firstgame-with-unity-and-csharp-#:~:text=Unity%20is%20a%20native%20C%20Mono%20or%20the%20Microsoft%20magazine/2014/august/unity-developing-your-first-game-with-unity-and-csharp-#:~:text=Unity%20is%20a%20native%20C%20Mono%20or%20the%20Microsoft%20developing-your-first-game-with-unity-and-csharp#:~:text=Unity%20is%20a%20native%20C%20Mono%20or%20the%20Microsoft%20M%20icrosoft%20.)