VISVESVARAYA TECHNOLOGICAL UNIVERSITY
JNANA SANGAMA, BELGAUM- 590014

Internship Report On

"AI BASED GAMING ENGINE"

Submitted in partial fulfillment for the requirements of the VII Semester degree of

BACHELOR OF ENGINEERING
IN
COMPUTER SCIENCE AND ENGINEERING

For The Academic Year
2022-23

SUBMITTED BY:-

AMOD KUMAR JHA	**CHETHAN S**	**GURUPRASAD S R**
1DB19CS009	**1DB19CS035**	**1DB19CS056**

Internship Carried Out At

AUTOMATA RESEARCH LABORATORIES

Project Guide	**Project Coordinator**
MR. RAGHAVENDRA SWAMY H	**DR. VENUGEETHA Y**
Founder and Proprietor	Professor, Dept. of CSE
Automata Research Laboratories	DBIT, Bangalore

DON BOSCO INSTITUTE OF TECHNOLOGY, BENGALURU-560074

DON BOSCO INSTITUTE OF TECHNOLOGY

Kumbalagodu, Bangaluru – 560074

DECLARATION

We, AMOD KUMAR JHA, CHETHAN S and GURUPRASAD S R, student of seven semester B.E, Department of Computer Science and Engineering, Don Bosco Institute of Technology, Kumbalagodu, Bengaluru, declare, that the internship work entitled **"AI BASED GAMING ENGINE"** has been carried out by me and submitted in partial fulfillment of the internship requirements for the award of degree in **Bachelor of Engineering in Computer Science and Engineering of Visvesvaraya Technological University**, Belgaum during the academic year 2022-23. The matter embodied in this report has not been submitted to any university or institute for the award of any other degree or diploma.

Place: Bengaluru	**AMOD KUMAR JHA**	**[1DB19CS009]**
	CHETHAN S	**[1DB19CS035]**
Date:	**GURUPRASAD S R**	**[1DB19CS056]**

DON BOSCO INSTITUTE OF TECHNOLOGY

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the internship project report entitled “**AI BASED GAMING ENGINE**” is a work carried out by AMOD KUMAR JHA (1DB19CS009), CHETHAN S (1DB19CS035) & GURUPRASAD S R (1DB19CS056) in partial fulfillment of award of **Degree of Bachelor of Engineering in Computer Science and Engineering** of Visvesvaraya Technological University, Belagavi, during the academic year 2022-2023. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated. The internship project has been approved as it satisfies the academic requirements associated with the degree mentioned.

Signature of Guide	Signature of HOD
Prof. Umashankar B S	**Prof. KB Shivakumar**
Vice Principal,	Head of Department
Dept. of CSE	Dept. of CSE
DBIT, Bengaluru.	DBIT, Bengaluru

Principal
Dr. B S Nagabhushana
Principal,
DBIT, Bengaluru

Name of External Examiners	**Signature with Date**
1.____________________	**1.** _________________
2.____________________	**2.**_________________

ABSTRACT

This internship helps us to understand the use and importance of artificial intelligence in the implementation of gaming engines. It aims to develop an AI-based gaming engine using Python and its related frameworks and libraries. The goal of our project is to design a python-based gaming application of the game “Stickman Fight” from scratch. For our project, we decided to design a 2D Plat former game where the objective of the game is to fight against each other. Twoplayers are given two different sets of keys to control their character. They have to bring the life points of the other user to end the game. The game is designed in a windows environment and written in python using PyCharm as IDE, integrated with pygame libraries. For our project, we have implemented several programming modules as discussed and designed in the course of the internship. As a result, we have created a 2-D game that is simple to understand and fun to play.

ACKNOWLEDGEMENT

Hereby we are submitting the Internship report on **“AI BASED GAMING ENGINE”,** as per the scheme of Visvesvaraya Technological University, Belgaum.

In this connection, we would like to express my deep sense of gratitude to my beloved institution Don Bosco Institute of Engineering and also, we would like to express our sincere gratitude and indebtedness to
Dr. B S Nagabhushana, Principal, DBIT, Bangalore.

We would like to express our sincere gratitude to **Dr. K B Shivakumar** Professor and Head of Dept. of Computer Science and Engineering, for providing a congenial environment to work in and carryout my internship.

We consider it my cardinal duty to express the deepest sense of gratitude to thank my Internship Project Guide **Mr. Raghavendra Swamy H,** Founder and Propritory at **Automata Research Laboratories** for his constant help and support extended towards me during the course of the project.

Finally, we are very much thankful to all the teaching and non teaching members of the Department of Computer Science and Engineering, our seniors, friends and our parents for their constant encouragement, support and help throughout completion of report.

AMOD KUMAR JHA	**[1DB19CS009]**
CHETHAN S	**[1DB19CS035]**
GURUPRASAD S R	**[1DB19CS056]**

CONTENTS

Chapter 1

INTRODUCTION

1.1 Python Programming Language

Python is a high-level, general-purpose programming language. Its design philosophy emphasizes code readability with the use of significant indentation.

Python is dynamically-typed and garbage-collected. It supports multiple programming paradigms, including structured (particularly procedural), object-oriented and functional programming. It is often described as a "batteries included" language due to its comprehensive standard library.

Guido van Rossum began working on Python in the late 1980s as a successor to the ABC programming language and first released it in 1991 as Python 0.9.0. Python 2.0 was released in 2000 and introduced new features such as list comprehensions, cycle-detecting garbage collection, reference counting, and Unicode support. Python 3.0, released in 2008, was a major revision that is not completely backward-compatible with earlier versions. Python 2 was discontinued in version 2.7.18 in 2020.

Python consistently ranks as one of the most popular programming languages.

Why Python?

Python has multiple Libraries and Frameworks most popular libraries are TensorFlow, Scikit-Learn, NumPy, Keras, Theano, and Pandas. Python is used in Big Data and Machine Learning. Python is used in Web Development's most popular sites in the world like Spotify, Instagram, Pinterest, Mozilla Firefox, Yelp, etc.

Python's popularity has risen dramatically in recent years and shows no signs of slowing. According to the Web, the programming language Python ranks second in popularity. Python has seen impressive growth of about 50 percent in the last year. We anticipate that Python's ranking will rise by another 50 percent in 2022.

1.2 Artificial Intelligence

Artificial intelligence (AI) refers to the simulation of human intelligence in machines that are programmed to think like humans and mimic their actions. The term may also be applied to any machine that exhibits traits associated with a human mind such as learning and problem-solving. The main characteristics of AI are autonomy and adaptivity.

Autonomy: The ability to perform tasks in complex environments without constant guidance by a user.

Adaptivity: The ability to improve performance by learning from experience.

Examples: self-driving cars, content recommendation (Netflix), GPS navigation systems for providing suggestions on the best route, and smart assistants like Siri and Alexa.

The ideal characteristic of artificial intelligence is its ability to rationalize and take actions that have the best chance of achieving a specific goal. A subset of artificial intelligence is machine learning (ML), which refers to the concept that computer programs can automatically learn from and adapt to new data without being assisted by humans. Deep learning techniques enable this automatic learning through the absorption of huge amounts of unstructured data such as text, images, or video.

1.3 AI-based Gaming Engine

Game Engine is a software framework primarily designed for the development of video games, and generally includes relevant libraries and support programs.[1] The "engine" terminology is similar to the term "software engine" used in the software industry. Game engine can also refer to the development software utilizing this framework, typically offering a suite of tools and features for developing games. Artificial Intelligence is used to generate responsive, adaptive or intelligent behaviours primarily in non-player characters (NPCs) similar to human-like intelligence. Artificial intelligence has been an integral part of video games since their inception in the 1950s. AI in video games is a distinct subfield and differs from academic AI. It serves to improve the game-player experience rather than machine learning or decision making.

AI in gaming refers to responsive and adaptive video game experiences. These AI-powered interactive experiences are usually generated via non-player characters, or NPCs, that act intelligently or creatively, as if controlled by a human game player. AI is the engine that determines an NPC’s behavior in the game world.While AI in some form has long appeared in video games, it is considered a booming new frontier in how games are both developed and played. AI games increasingly shift the control of the game experience toward the player, whose behavior helps produce the game experience. AI procedural generation, also known as procedural storytelling, in game design refers to game data being produced algorithmically rather than every element being built specifically by a developer.

A game engine is a software development program or environment used to develop video games. The features of a game engine may include artificial intelligence. In video games, artificial intelligence (AI) is used to generate responsive, adaptive, or intelligent behaviors similar to human-like intelligence. Examples: Amazon Lumberyard, CryEngine 3, Panda 3D, Unity 3D, Unreal Engine.

1.4 Compiler

The AI compiler is the execution system for the super.AI assembly line described on our Data programming page.

It ensures that all inputs are split into smaller tasks that are efficiently and accurately labelled before being recombined into a coherent output. The AI compiler also uses the processed outputs it receives as training data for machine learning algorithms that go on to form an additional labelling source.

1.5 Machine Learning(ML)

Machine learning (ML) is a type of artificial intelligence (AI) that allows software applications to become more accurate at understanding and building methods that 'learn', that is, methods that leverage data to improve performance on some set of tasks. It is seen as a part of artificial intelligence. Machine learning is the method to train a computer to learn from its inputs but without explicit programming for every circumstance. Machine learning helps a computer to achieve artificial intelligence predicting outcomes without being explicitly programmed to do so Machine learning is a field of inquiry devoted to.

1.6 AI Data Analytics

The main responsibility of an AI Data Analyst includes procuring, preparing, cleansing and modelling data using machine learning models and new analytical methods. Also, the AI Data Analyst is responsible for Designing and creating data reports to help stakeholders to make better decisions

Chapter 2

INTERNSHIP WORKFLOW

2.1 MEETING OF MINUTES

MEETING DATE	MEETING MINUTES	DISCUSSION
25-08-22 07:00 – 07:10	10 MINS	• FINALIZED PROJECT TOPIC –AI based gaming engine. • Discussed about AI, gaming engine and, Compiler stack.
27-08-22 07:00 – 07:10	10 MINS	• FINALIZED PROJECT TOPIC AGAIN- AI Based Gaming Engine. • Finalized Which Operating System To Be Used- WINDOWS • Discussed About Python Latest Version.
29-08-22 07:00 – 07:10	10 MINS	• Installation of Python in each of our laptops • Writing of a simple "Hello World" program • Listing of Popular games • Listing of Popular Gaming Engines • Listing of AI based applications
30-08-22 07:00 – 07:10	10 MINS	• Finallised The Game Genre – Fighting/Shooting • Storyline Of The Game
01-09-22 07:00 – 07:10	10 MINS	• Background Finalization

		• Discussed About Python Ai Based Libraries
02-09-22 07:00 – 07:10	10 MINS	• Installation of python based AI libraries
03-09-22 07:00 – 07:10	10 MINS	• Discussion on the basic functionalities of characters and background.
05-09-22 07:00 – 07:10	10 MINS	• To finalize the game characters and the background
06-09-22 07:00 – 07:10	10 MINS	• Editing and finalizing the character design and the background
08-09-22 07:00 – 07:10	10 MINS	• Discussion unified modeling language significance and its designing tools.
10-09-22 07:00 – 07:10	10 MINS	• Discussion relating Algorithm And FlowChart for the Game
12-09-22 07:00 – 07:10	10 MINS	• Modification of the game design.
14-09-22 07:00 – 07:10	10 MINS	• Discussion of the pygame API Functions
15-09-22 07:00 – 07:10	10 MINS	• Discussion of scikit learn and LIBSVM library

19-09-22 07:00 – 07:20	20 MINS	• Implementing Game Menu
21-09-22 07:00 – 07:10	10 MINS	• Discussion on Placement & Interview
23-09-22 07:00 – 07:15	15 MINS	• Talked about project and all
25-09-22 11:00 – 11:45	45 MINS	• Discussed on different types of software licenses

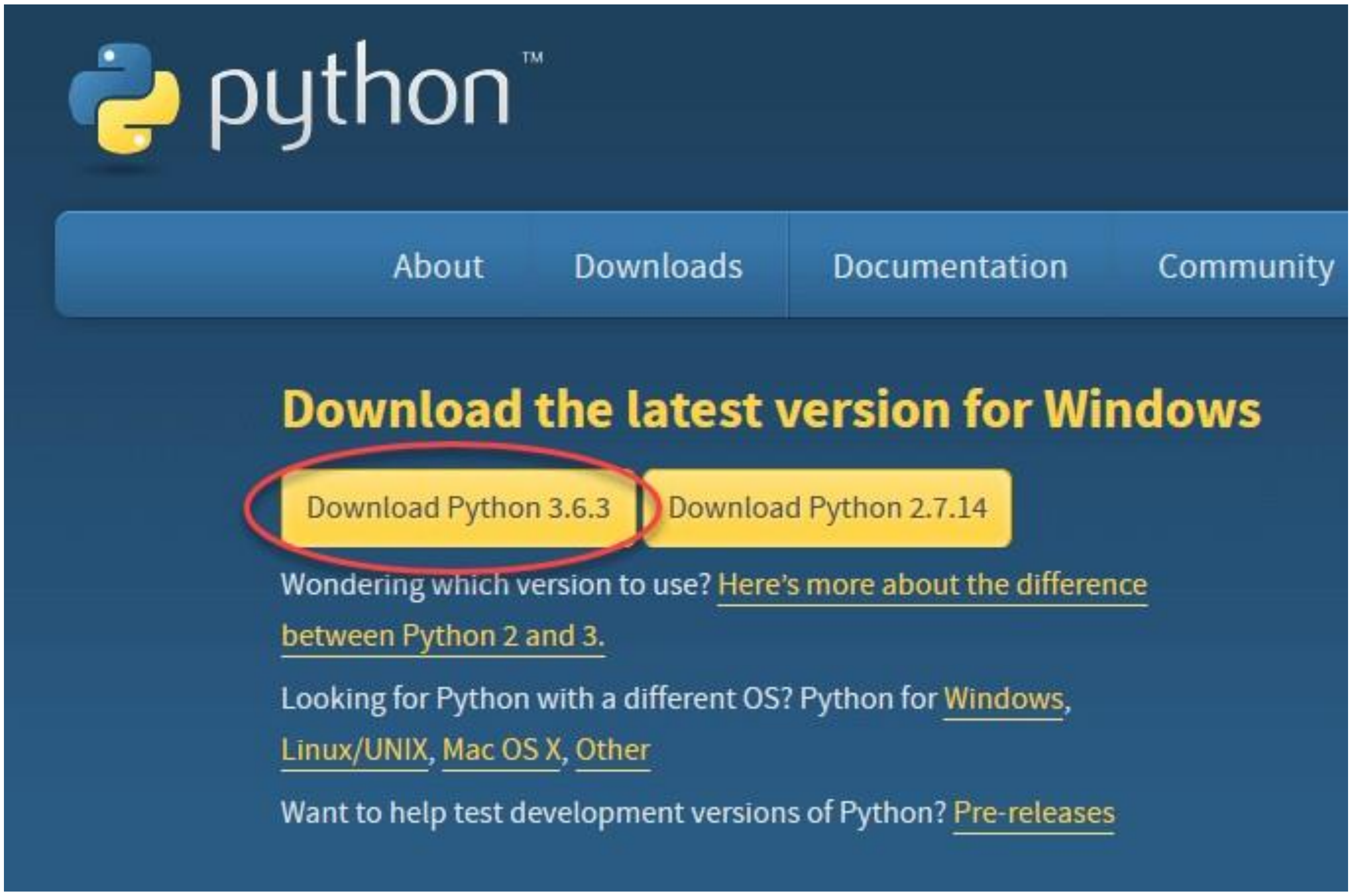
Chapter 3

SOFTWARE INSTALLATIONS

3.1 Installation Steps of Python IDLE

Here is a step by step process on how to download and install Python IDLE on Windows:

STEP 1) To download and install Python, visit the official website of Python https://www.python.org/downloads/ and choose your version. We have chosen Python version 3.6.3



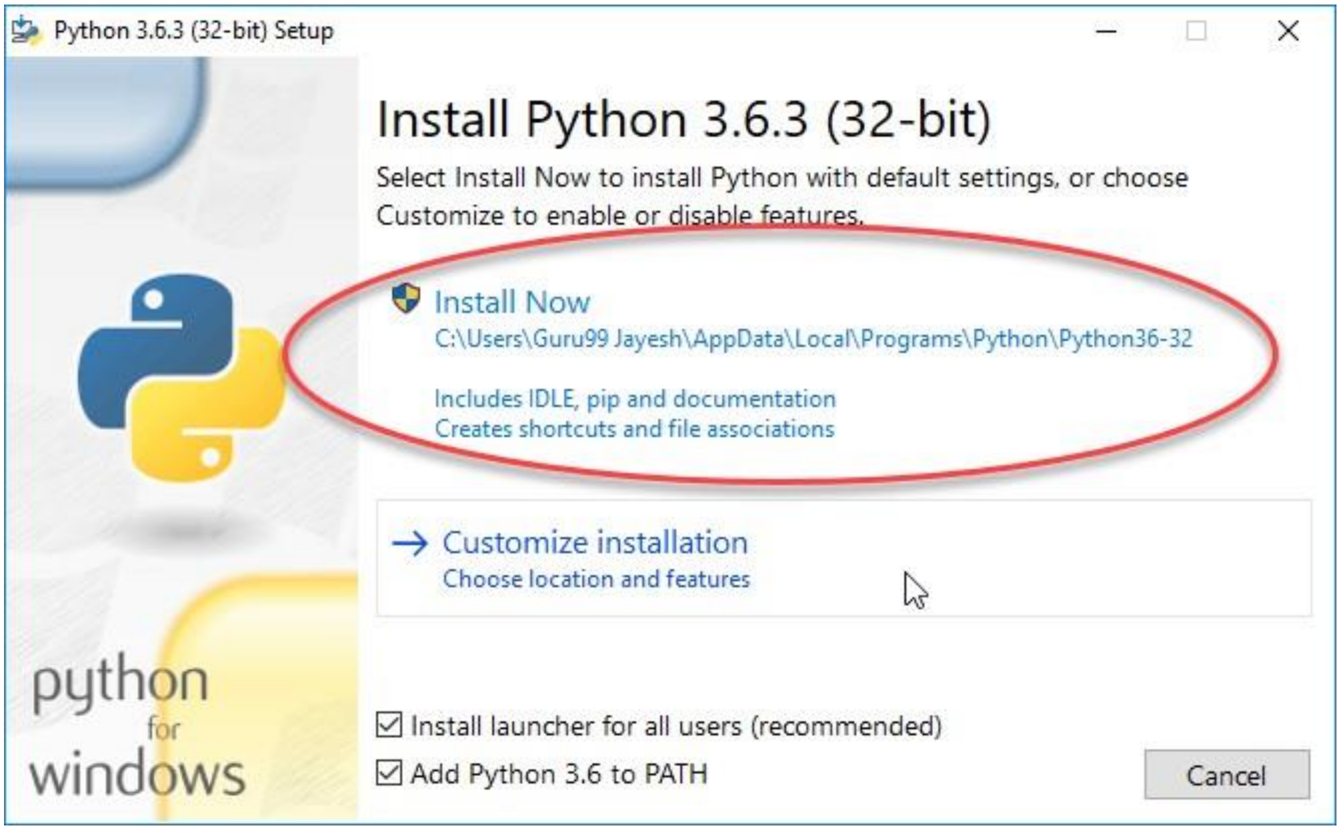
Download the latest version for Windows

Wondering which version to use? Here's more about the difference between Python 2 and 3.

Looking for Python with a different OS? Python for Windows, Linux/UNIX, Mac OS X, Other

Want to help test development versions of Python? Pre-releases

Step 2) Once the download is completed, run the .exe file to install Python. Now click on Install Now.


Python 3.6.3 (32-bit) Setup

Install Python 3.6.3 (32-bit)

Select Install Now to install Python with default settings, or choose Customize to enable or disable features.

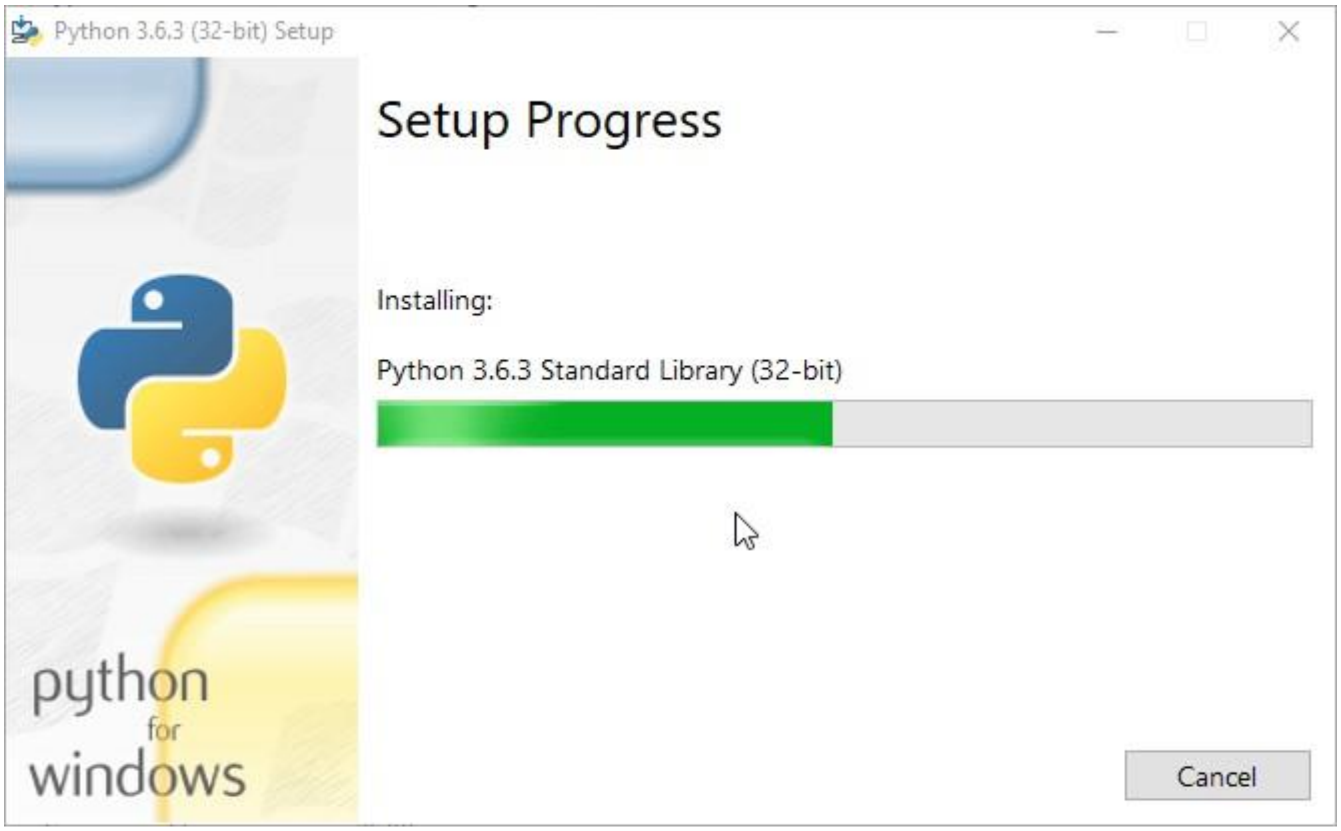
Install Now
C:\Users\Guru99 Jayesh\AppData\Local\Programs\Python\Python36-32

Includes IDLE, pip and documentation
Creates shortcuts and file associations

→ Customize installation
Choose location and features

☑ Install launcher for all users (recommended)
☑ Add Python 3.6 to PATH

Step 3) You can see Python installing at this point.

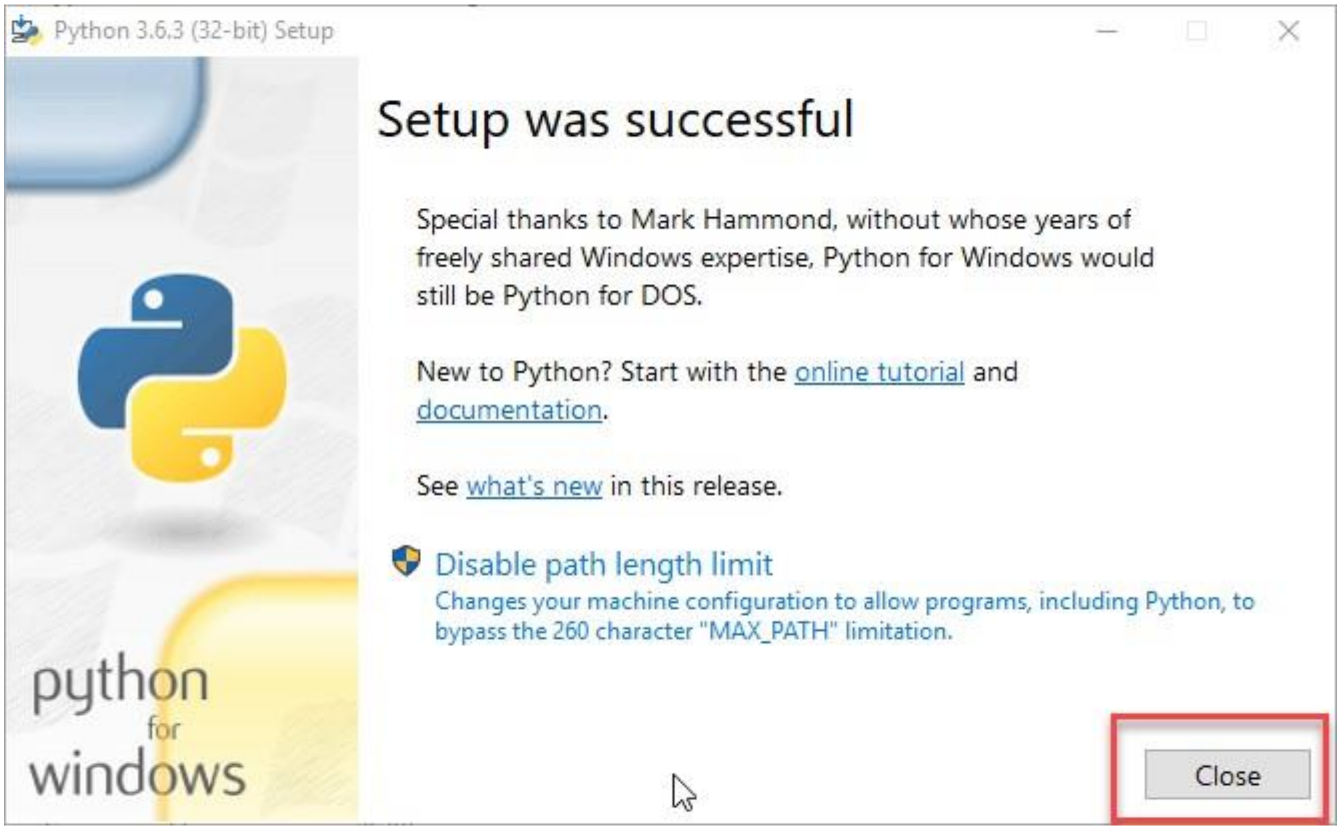

Python 3.6.3 (32-bit) Setup

Setup Progress

Installing:

Python 3.6.3 Standard Library (32-bit)

Step 4) When it finishes, you can see a screen that says the Setup was successful. Now click on “Close”.



Python 3.6.3 (32-bit) Setup

Setup was successful

Special thanks to Mark Hammond, without whose years of freely shared Windows expertise, Python for Windows would still be Python for DOS.

New to Python? Start with the online tutorial and documentation.

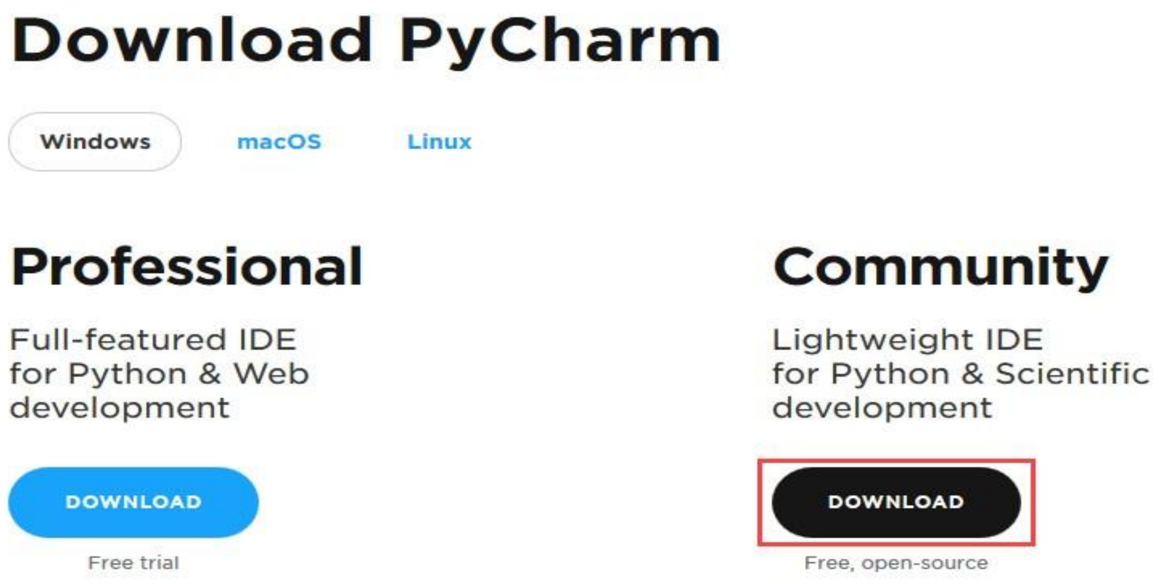
See what's new in this release.

Disable path length limit
Changes your machine configuration to allow programs, including Python, to bypass the 260 character "MAX_PATH" limitation.

3.2 Installation Steps of Pycharm:-

Here is a step by step process on how to download and install Pycharm IDE on Windows:

Step 1) To download Pycharm visit the website https://www.jetbrains.com/pycharm/download/ and Click “DOWNLOAD” link under the Community Section.

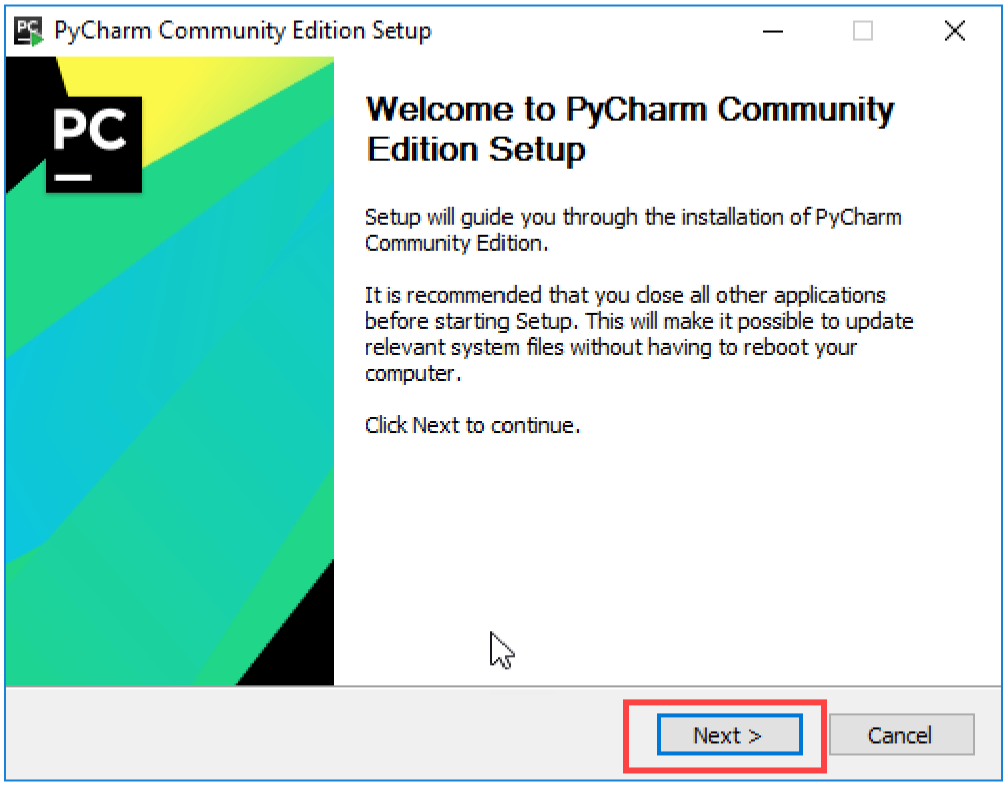


Download PyCharm

Professional
Full-featured IDE for Python & Web development

Community
Lightweight IDE for Python & Scientific development

Step 2) Once the download is complete, run the exe for install PyCharm. The setup wizard should have started. Click "Next".



PyCharm Community Edition Setup

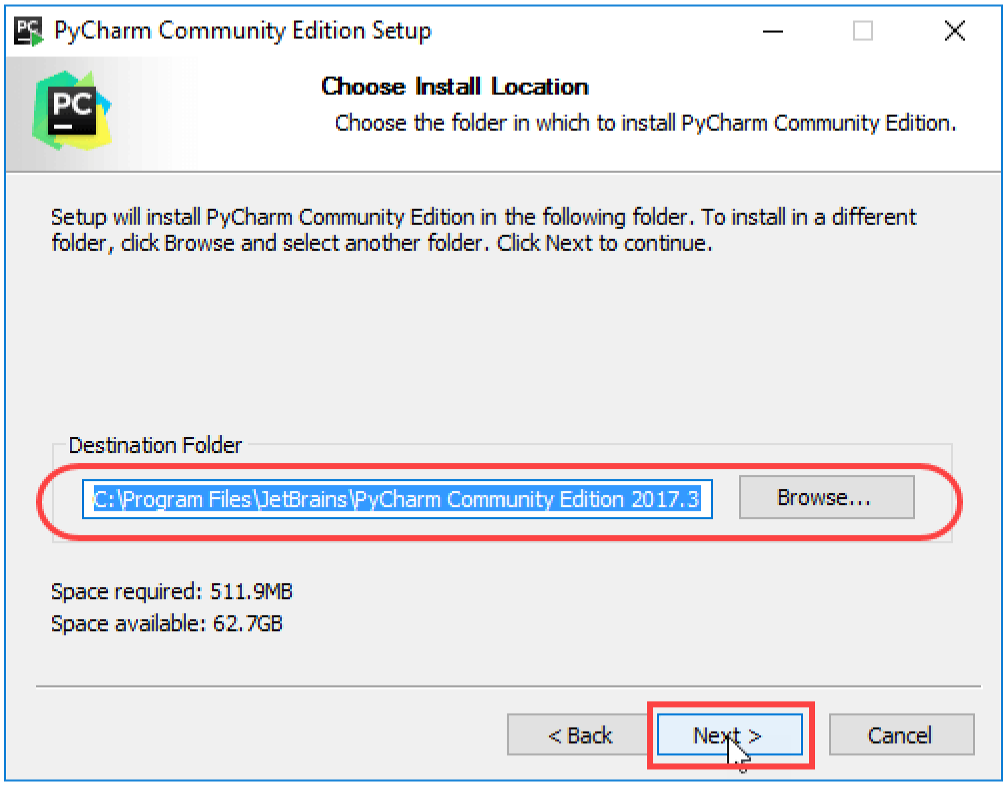
Welcome to PyCharm Community Edition Setup

Setup will guide you through the installation of PyCharm Community Edition.

It is recommended that you close all other applications before starting Setup. This will make it possible to update relevant system files without having to reboot your computer.

Click Next to continue.

Step 3) On the next screen, Change the installation path if required. Click "Next".



PyCharm Community Edition Setup

Choose Install Location

Choose the folder in which to install PyCharm Community Edition.

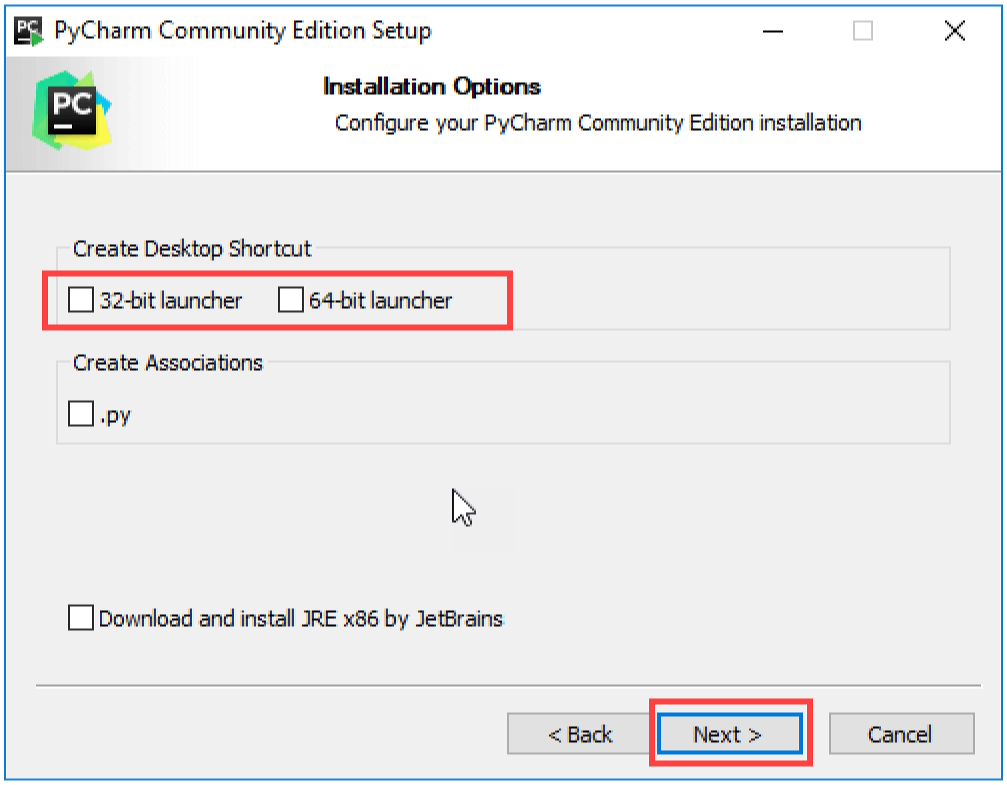
Setup will install PyCharm Community Edition in the following folder. To install in a different folder, click Browse and select another folder. Click Next to continue.

Destination Folder

C:\Program Files\JetBrains\PyCharm Community Edition 2017.3

Space required: 511.9MB
Space available: 62.7GB

Step 4) On the next screen, you can create a desktop shortcut if you want and click on "Next".



PyCharm Community Edition Setup

Installation Options

Configure your PyCharm Community Edition installation

Create Desktop Shortcut

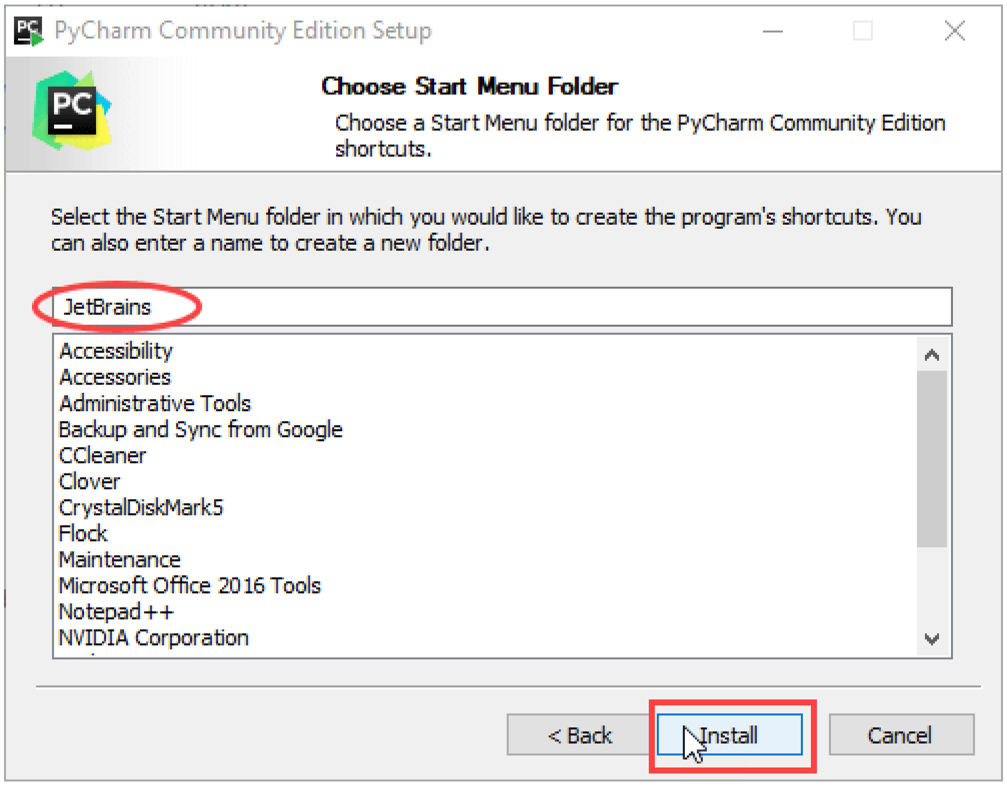
☐ 32-bit launcher ☐ 64-bit launcher

Create Associations

☐ .py

☐ Download and install JRE x86 by JetBrains

Step 5) Choose the start menu folder. Keep selected JetBrains and click on "Install".



PyCharm Community Edition Setup

Choose Start Menu Folder

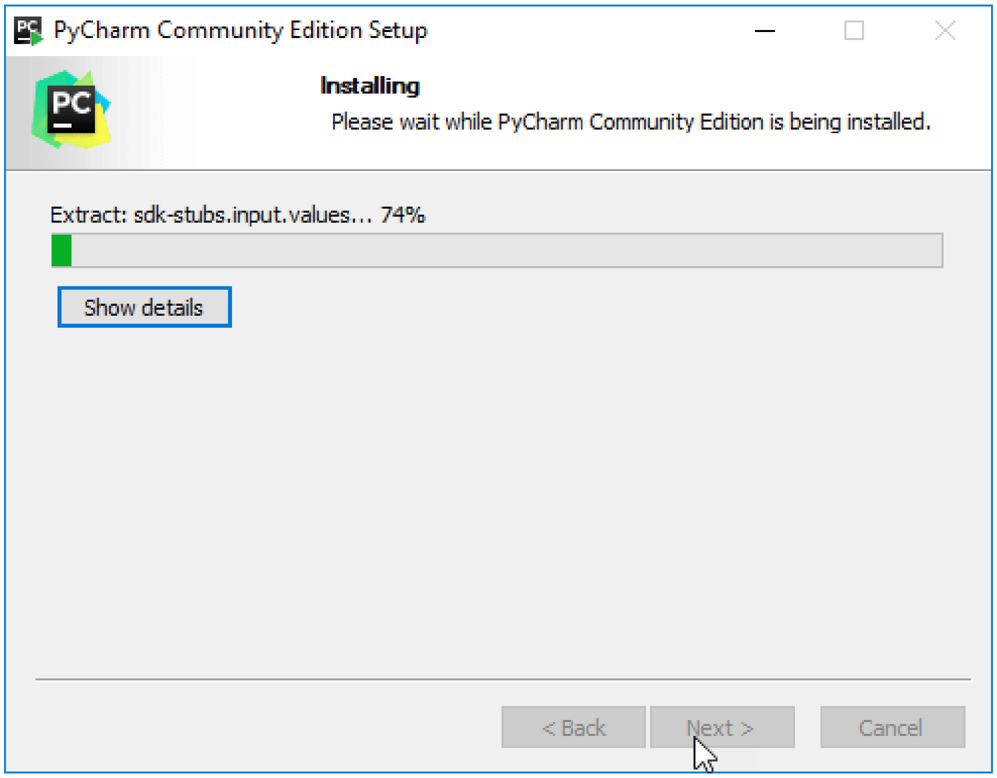
Choose a Start Menu folder for the PyCharm Community Edition shortcuts.

Select the Start Menu folder in which you would like to create the program's shortcuts. You can also enter a name to create a new folder.

JetBrains

Accessibility
Accessories
Administrative Tools
Backup and Sync from Google
CCleaner
Clover
CrystalDiskMark5
Flock
Maintenance
Microsoft Office 2016 Tools
Notepad++
NVIDIA Corporation

Step 6) Wait for the installation to finish.



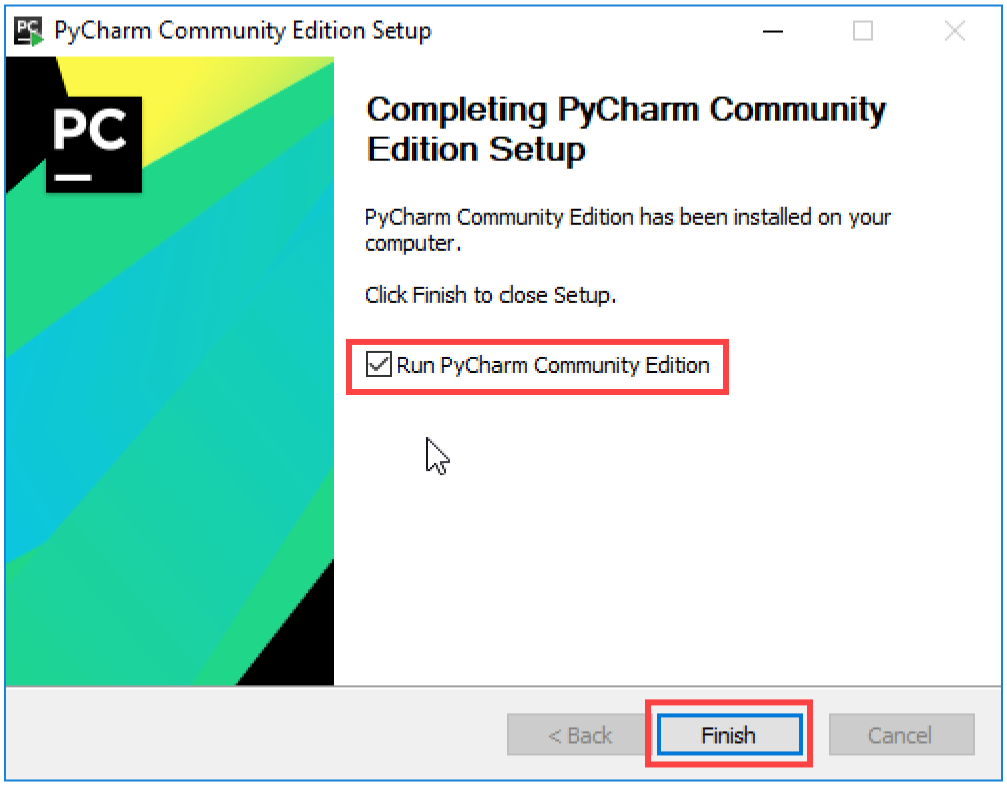
PyCharm Community Edition Setup

Installing

Please wait while PyCharm Community Edition is being installed.

Extract: sdk-stubs.input.values... 74%

Step 7) Once installation finished, you should receive a message screen that PyCharm is installed. If you want to go ahead and run it, click the "Run PyCharm Community Edition" box first and click "Finish".



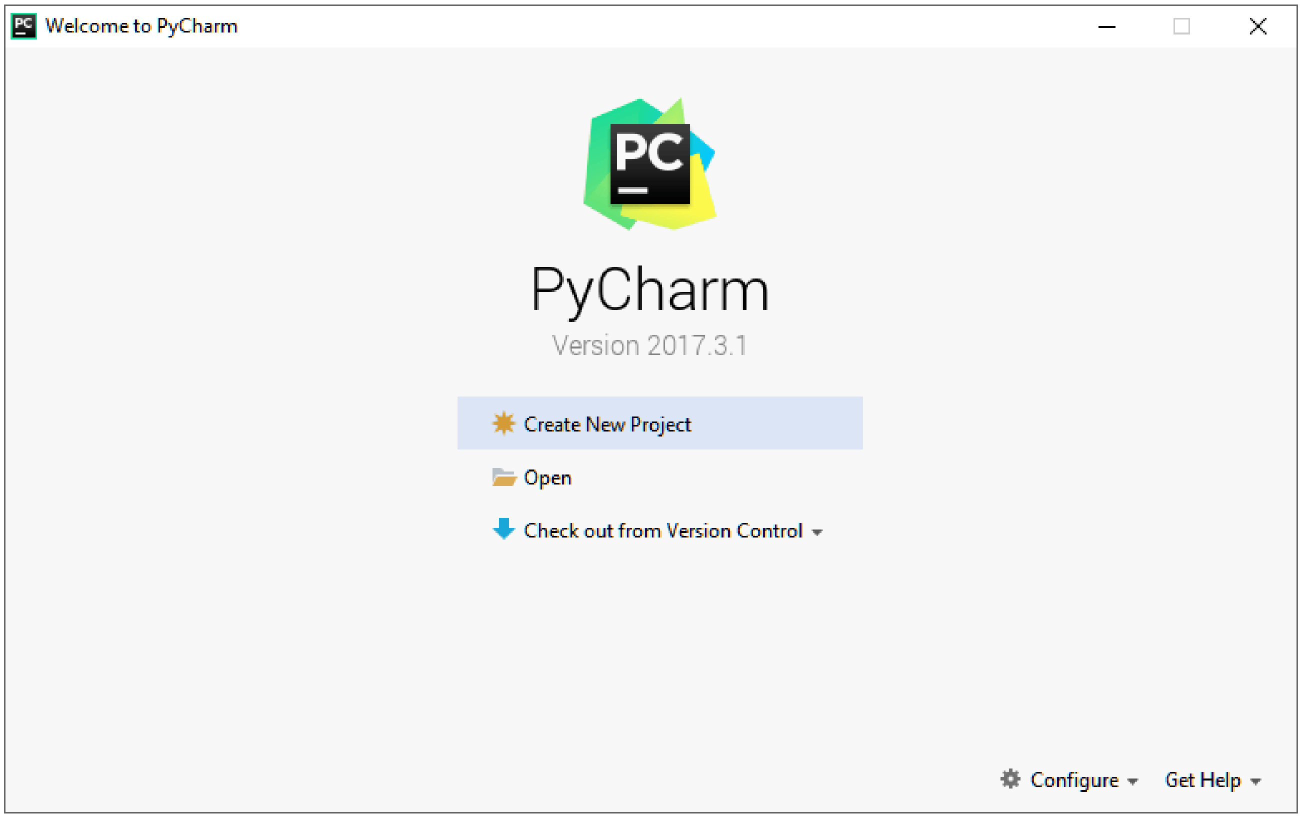
Completing PyCharm Community Edition Setup

PyCharm Community Edition has been installed on your computer.

Click Finish to close Setup.

[✓] Run PyCharm Community Edition

Step 8) After you click on “Finish,” the Following screen will appear.



Chapter 4

Software Development Life Cycle (SDLC) steps

Defining SDLC :- A systematic approach that generates a structure for the developer to design, create and deliver high-quality software based on customer requirements and needs. The primary goal of the SDLC process is to produce cost-efficient and high-quality products. The process comprises a detailed plan that describes how to develop, maintain, and replace the software.

Phases Of SDLC :-

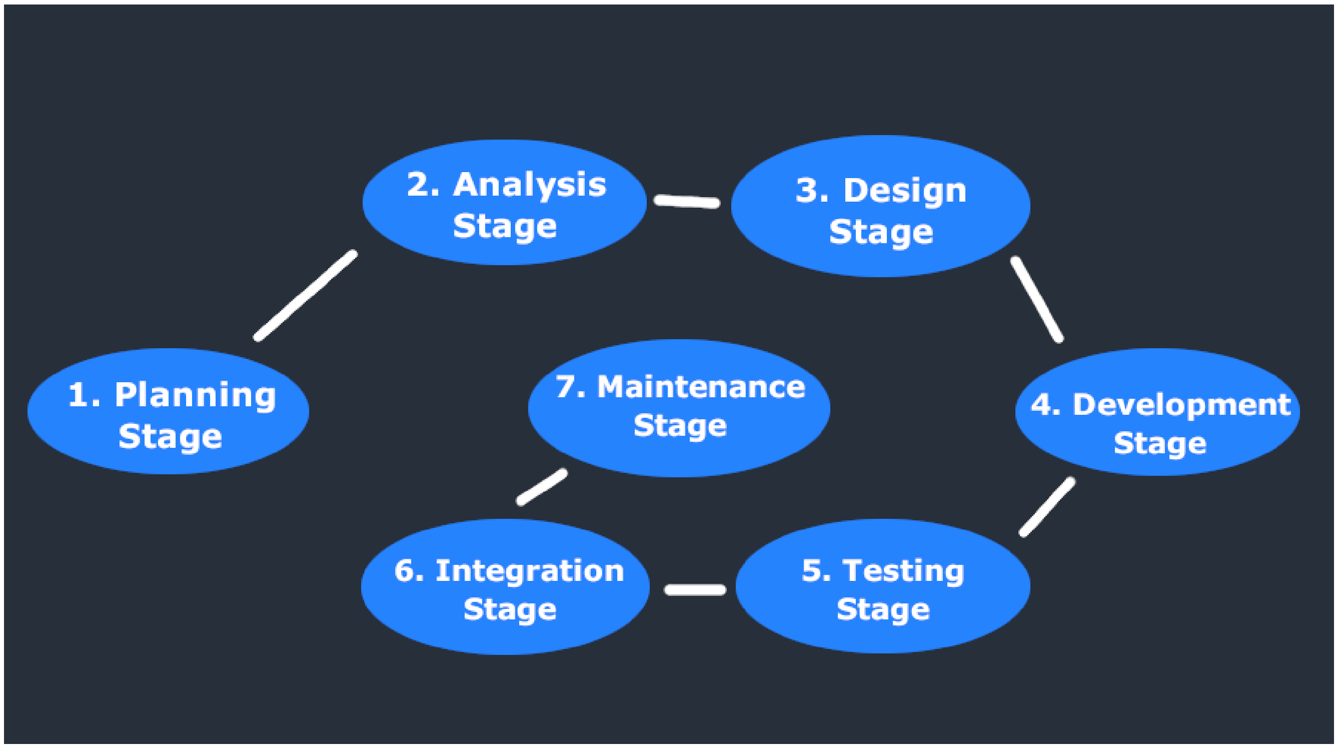


fig : Software Development Life Cycle Steps

1: Project Planning

The first stage of SDLC is all about "What do we want?" Project planning is a vital role in the software delivery lifecycle since this is the part where the team estimates the cost and defines the requirements of the new software.

2: Gathering Requirements & Analysis

The second step of SDLC is gathering maximum information from the client requirements for the product. Discuss each detail and specification of the product with the customer. The development team will then analyze the requirements keeping the design and code of the software in mind. Further, investigating the validity and possibility of incorporating these requirements into the software system. The main goal of this stage is that everyone understands even the minute detail of the requirement. Hardware, operating systems, programming, and security are to name the few requirements

3: Design

In the design phase (3rd step of SDLC), the program developer scrutinizes whether the prepared software suffices all the requirements of the end-user. Additionally, if the project is feasible for the customer technologically, practically, and financially. Once the developer decides on the best design approach, he then selects the program languages like Oracle, Java, etc., that will suit the software.

Once the design specification is prepared, all the stakeholders will review this plan and provide their feedback and suggestions. It is absolutely mandatory to collect and incorporate stakeholder's input in the document, as a small mistake can lead to cost overrun.

4: Coding or Implementation

Time to code! It means translating the design to a computer-legible language. In this fourth stage of SDLC, the tasks are divided into modules or units and assigned to various developers. The developers will then start building the entire system by writing code using the programming languages they chose. This stage is considered to be one of the longest in SDLC. The developers need certain predefined coding guidelines, and programming tools like interpreters, compilers, debugger to implement the code.

The developers can show the work done to the business analysts in case if any modifications or enhancements required.

5: Testing

Once the developers build the software, then it is deployed in the testing environment. Then the testing team tests the functionality of the entire system. In this fifth phase of SDLC, the testing is done to ensure that the entire application works according to the customer requirements.

After testing, the QA and testing team might find some bugs or defects and communicate the same with the developers. The development team then fixes the bugs and send it to QA for a re-test. This process goes on until the software is stable, bug-free and working according to the business requirements of that system.

6: Deployment

The sixth phase of SDLC: Once the testing is done, and the product is ready for deployment, it is released for customers to use. The size of the project determines the complexity of the deployment. The users are then provided with the training or documentation that will help them to operate the software. Again, a small round of testing is performed on production to ensure environmental issues or any impact of the new release.

7: Maintenance

The actual problem starts when the customer actually starts using the developed system and those needs to be solved from time to time. Maintenance is the seventh phase of SDLC where the developed product is taken care of. According to the changing user end environment or technology, the software is updated timely.

Chapter 5

REQUIREMENT ANALYSIS

5.1 Characters Finalization

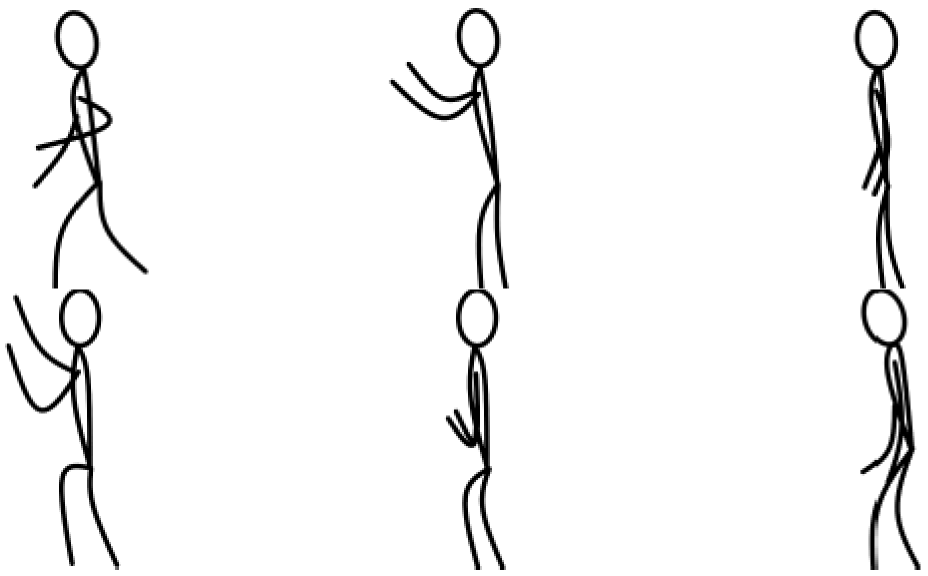


Fig : Left Jump

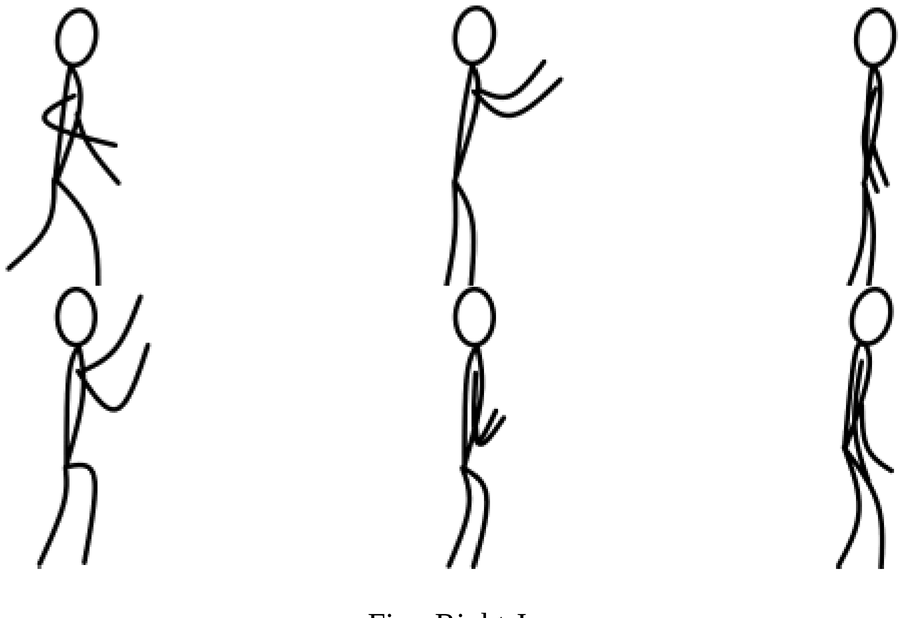


Fig : Right Jump

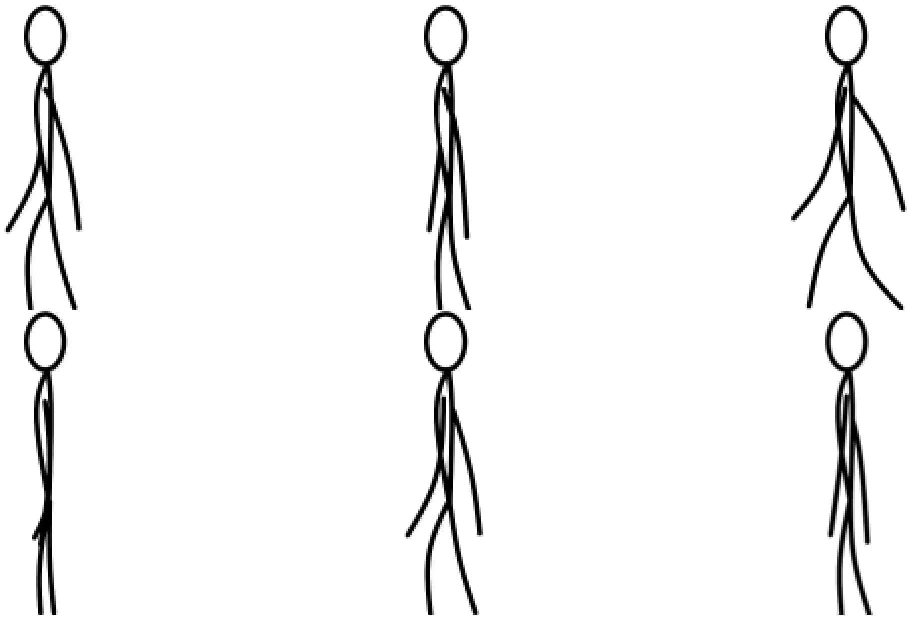


fig : Left Walking

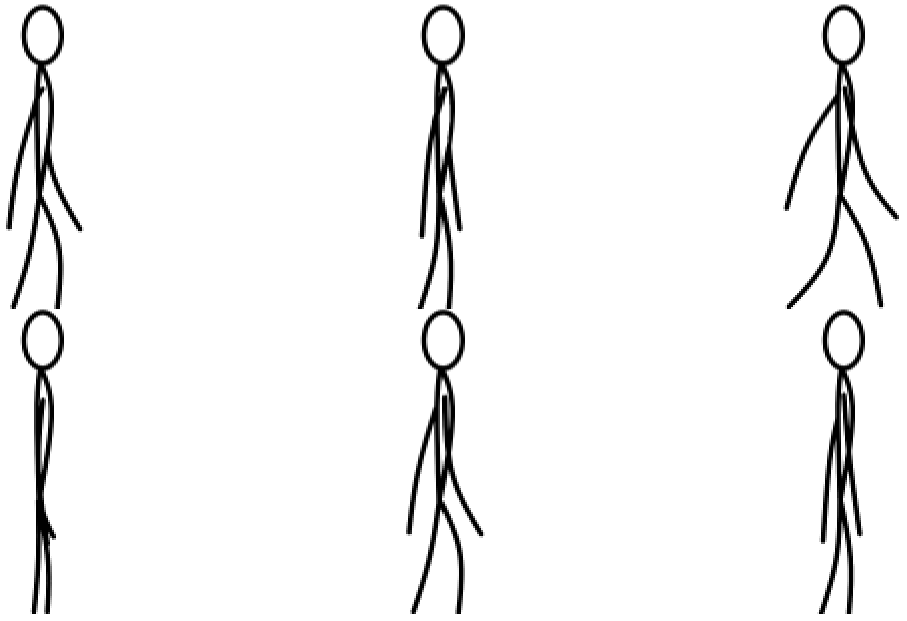


Fig : Right Walking

Fig : Enemy

Fig : Enemy Attack

5.2 Generation of Background

The background of the game is set in a place of temple region and also with a red background and a black platform to play the game.
The Warrior is towards the left side of the screen and the Wizard is towards the right side of the screen.

Fig : Background Image

5.3 Genre of Game Task

- Sandbox
  E.g.: Minecraft, Grand Theft Auto, The Sims
- Real-time strategy (RTS)
  Warcraft, Age of Empires, Command and Conquer
- Shooters (FPS and TPS)
  Halo, DOOM, Gear of War
- Multiplayer online battle arena (MOBA)
  Dota 2, Valorant, League of Legends

5.4 Python Based AI Libraries

1. Numpy

It is a popular Python library used to handle multi-dimensional data and complex mathematical functions being used on the data. Numpy library powers the speed of computation of mathematical expressions and execution of complex functions working on arrays.

Key Features:

- Shape manipulation.
- Discretion of Fourier transformations.
- Statistical operations and linear algebra.
- Support for n-dimensional arrays.
- Data cleaning and manipulation.
- Random simulations.

2. Pandas

Pandas is a prominent Python library generally used for Machine Learning concepts. It is basically a data analysis library that analyses and manipulates the data. Pandas make it easier for the developers to work with structured multidimensional data and time series concepts and produce efficient results.

Key Features:

- Data alignment and handling of missing data.
- Merging and joining of datasets.
- Dataset reshaping and pivoting.
- Data filtration.
- Data manipulation and analysis.
- Indexing of the data.

3. Matplotlib

It is a data visualization library used for designing plots and graphs. The library itself is an extension of SciPy and handles complex data models of Pandas as well as NumPy data structures. Matplotlib offer features such as Basemap, GTK tools, Cartopy, Mplot4d, etc., that help in generating image plots, 3D plots, contour plots, and more.

Key Features:

- High-quality diagrams, plots, histograms, graphs, etc.
- Intuitive and easy to use.
- GUI toolkit support.
- Map projections.
- Recognition of data patterns.

4. **SciPy**

It is a Python library that originates from NumPy. SciPy is leveraged by Python development services to perform technical and scientific computing on large sets of data. The library has ib-built array optimization and linear algebra modules that help in scientific analysis and engineering.

Key Features:

- Array manipulation subroutines.
- User-friendliness.
- Data visualization and manipulation.
- Scientific and technical analysis.

5. **Scikit-learn**

It is a powerful Python library that was originally generated to serve the purpose of data modeling and building machine learning algorithms. It has a simple, engaging, and consistent interface that is exceptionally user-friendly, making it easy to use and share data.

Key Features:

- Data modeling.
- End-to-end Machine Learning algorithms.
- Model selection.
- Classification of data.
- Dimensionality reduction.
- Pre-processing of the data.

5.5 Storyline Task2

- o Starting out with Forest and then continuing to Temple and then ending with Battlefield.
- o The main hero of the game will be a stick figure and each time the stickman will win a fight he will increase in size and increase in strength and weapons. The villain of the game is going to be a stickman too and having weapons that will be used.
- o The game is about a stick man who fights various villains to survive in the world of stick figures.
- o The game starts with basic villains with hand to hand combat and then advancing with various weapons. The stick man is a hero character which has basic skills while starting the game and then after defeating the enemies the hero gets to upgrade itself with strength and weapon choice.
- o The background of the game starts off with a forest area and then as the levels proceed the background also changes.
- o This game contains has finite levels and as the levels increase the number of enemies and also strength of them also increases.
- o There also contains a final boss fight where the stickman has to fight and has to do particular combos in order to defeat.
- o There is health bar for the stickman which denotes the health of the hero and once the health bar vanishes it means that the character is dead.
- o Throughout the game the hero is given 3 hearts which represents the lives the character has and once the hearts are also destroyed the game ends and the player needs to start the game from the beginning and will reset all the upgrades the player had made.
- o There are coins that can be collected and also the enemies also drop various coins which can be used to purchase various abilities and weapons and also increase the strength of the character.

5.6 UML (Unified Modeling Language)

Definition:- It is a general-purpose, developmental, modeling language in the field of software engineering that is intended to provide a standard way to visualize the design of a system.
It is a standard notation for the modeling of real world objects as a first step in developing an object-oriented design methodology.
The main aim of UML is to define a standard way to visualize the way a system has been designed. It is quite similar to blueprints used in other fields of engineering. UML is not a programming language, it is rather a visual language.

SIGNIFICANCE:-

- To communicate the desired structure and behavior of our system.
- To visualize and control the system's architecture.
- To better understand the system we are building.

- To often exposing opportunities for simplification and reuse and to manage risk.
- Provides template that guides us in constructing a system.
- Helps to understand complex system part by part.

UML Subtools :-

- Micro Focus Together
- Cacoo
- Creately
- Dia
- BOUML
- Edraw Max
- Enterprise Architect (software)
- Generic Modeling Environment
- JetUML
- MagicDraw
- Menthor Editor
- Microsoft Visio
- Modelio
- OntoUML
- OptimalJ
- PlantUML
- PowerDesignerProsa
- UML Modeller
- Rational Rhapsody
- IBM Rational Rose XDE
- Software Ideas Modeler
- StarUML
- UModel
- Visual Paradigm
- YEd

Chapter 6

DESIGN

6.1 Character Design

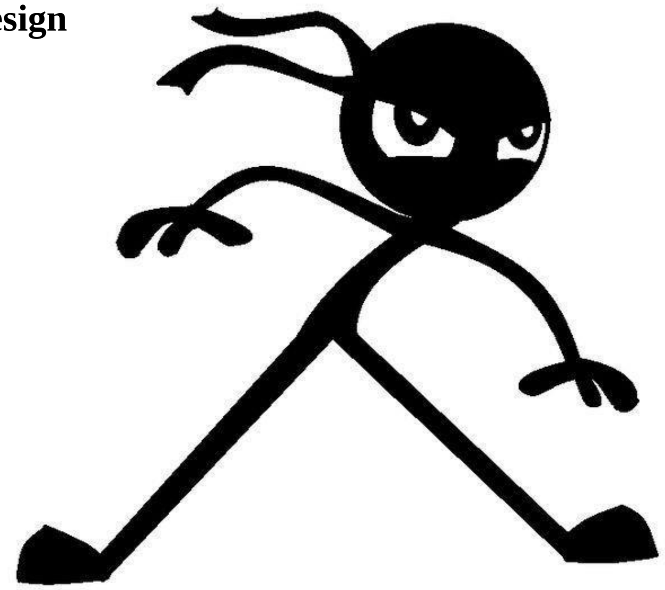


Fig : Stickman (Hero)

두건맨
(졸라맨) by. 김득헌
젠도 by. 그러
블랙연소
(샤이닝 파이터)
by. 펀플

Fig : Stickvillains (Villain)

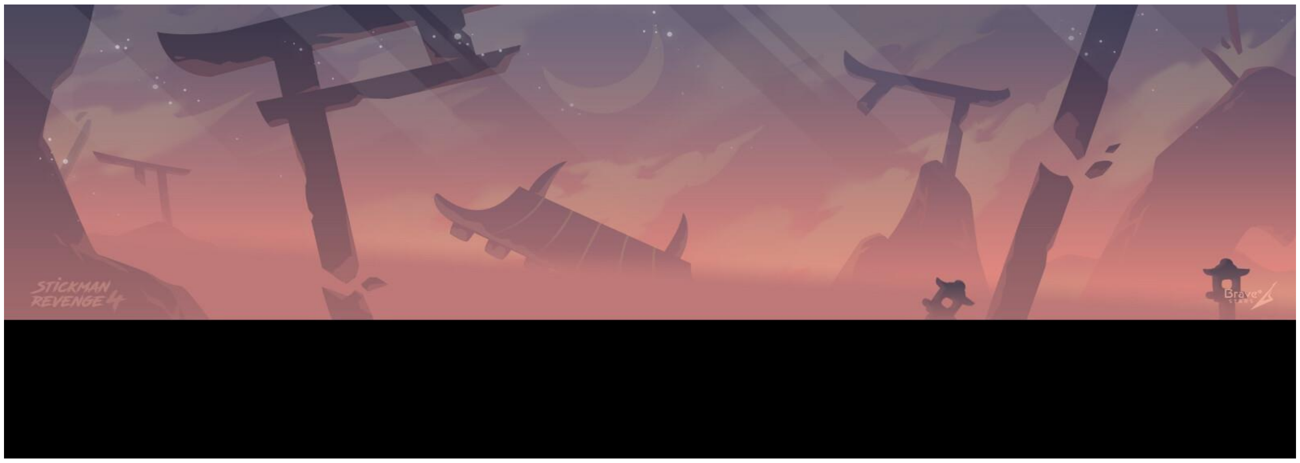


Fig : Forest Background

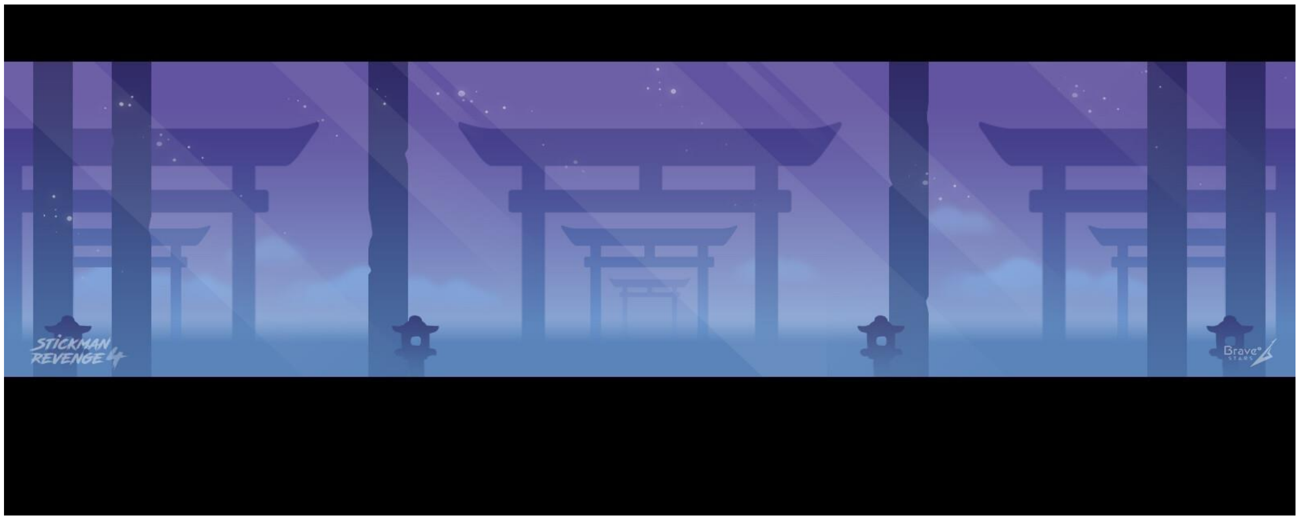


Fig : Temple Background

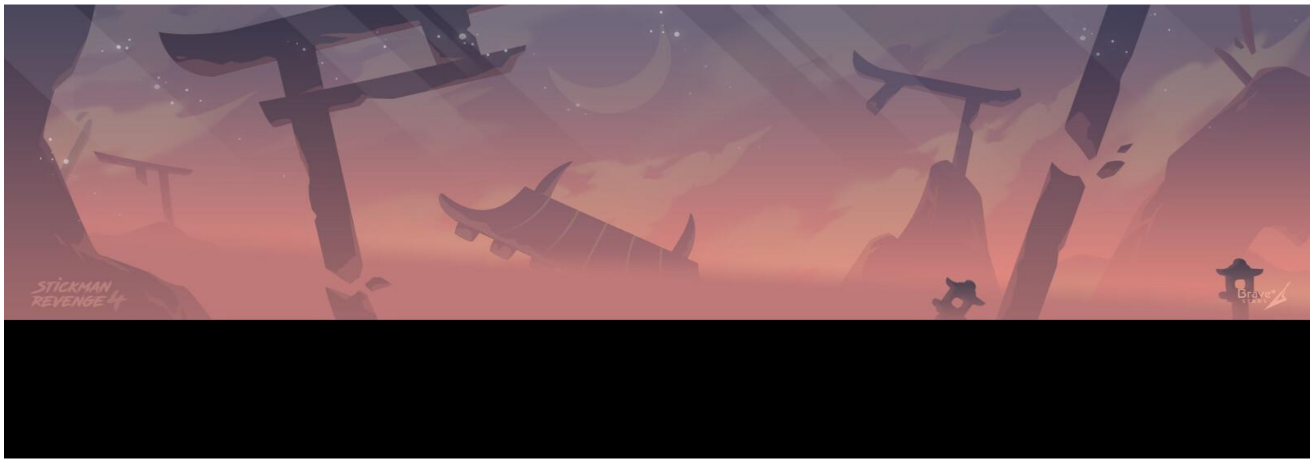


Fig : BattleField Background

Chapter 7

ALGORITHMS, FLOWCHARTS, AND PSEUDOCODES

7.1 Algorithm, flowchart & Pseudocode for game

ALGORITHM

```
STEP 1 :    Start
STEP 2 :    Initialize background
STEP 3 :    Initialize Warrior
STEP 4 :    Initialize Wizard
STEP 5:     Initialize the Health Bars and the weapons
STEP 6 :    While warrior or wizard is Alive, do Steps 7, 8.
STEP 7 :    Get user input to control the Warrior movement.
STEP 8 :    Get user input to control the Wizard movement.
STEP 9 :    If Warrior dies :
                          Print “WIZARD VICTORY”
STEP 10 :   If Wizard dies :
                          Print “WARRIOR VICTORY”
STEP 11 :   END
```

FLOWCHART :

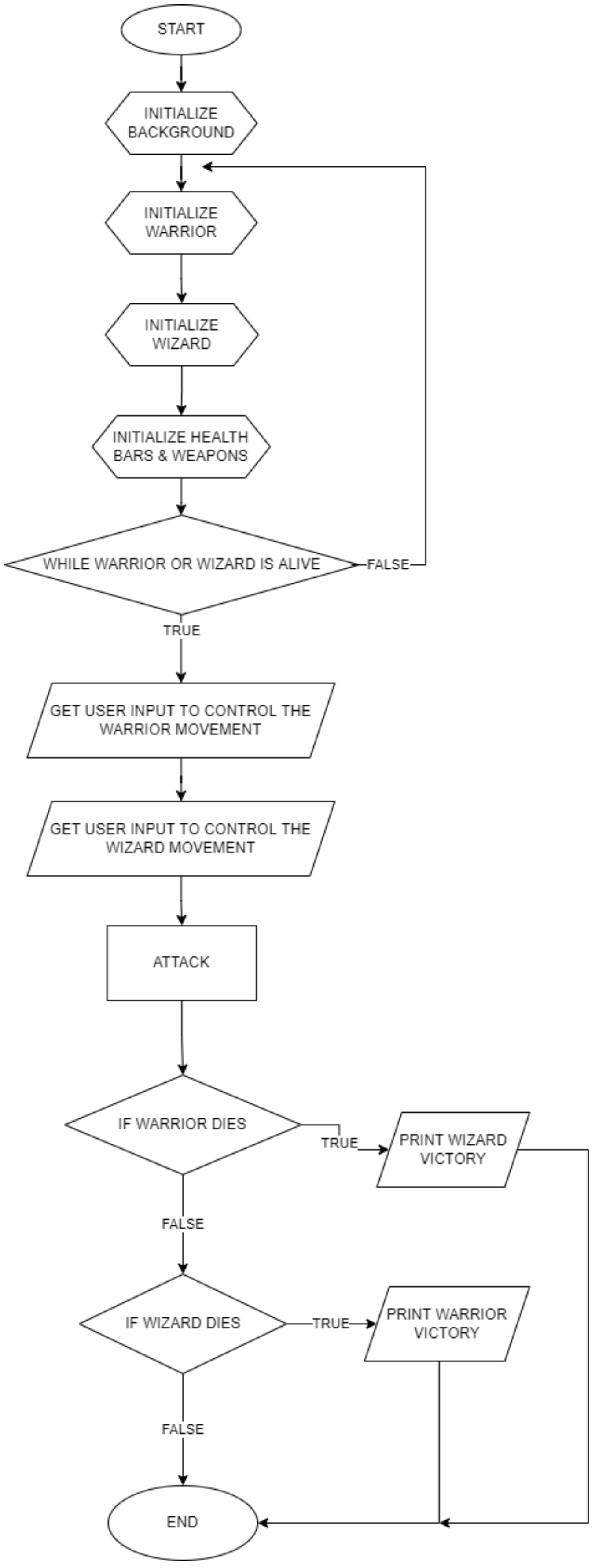

START
INITIALIZE BACKGROUND
INITIALIZE WARRIOR
INITIALIZE WIZARD
INITIALIZE HEALTH BARS & WEAPONS
WHILE WARRIOR OR WIZARD IS ALIVE
FALSE
TRUE
GET USER INPUT TO CONTROL THE WARRIOR MOVEMENT
GET USER INPUT TO CONTROL THE WIZARD MOVEMENT
ATTACK
IF WARRIOR DIES
TRUE
PRINT WIZARD VICTORY
FALSE
IF WIZARD DIES
TRUE
PRINT WARRIOR VICTORY
FALSE
END

Fig: Flowchart for the Game

PSEUDO CODE

```
While run:
Draw_BG()
        Draw_Warrior()
        Draw_Wizard()
        Draw_health bar()
WHILE Warrior or Wizard is ALIVE:
        Get_Warrior_input()
        Get_Wizard_input()
        continue
        If Warrior_Health_bar ==0:
                Print("WIZARD VICTORY")
        If Wizard_Health_bar==0:
                Print("WARRIOR VICTORY")
```

7.2 Algorithm, flowchart & Pseudocode for Warrior :

Algorithm

```
STEP 1 :    Start

STEP 2 :    If health bar of warrior == 0

                return wizard victory

            else

                continue

STEP 3 :    Get_Warrior_Input:

            If  Warrior _Input ==W:

                JUMP()

            Elif Warrior _Input ==A:

                MOVE_LEFT()

            Elif Warrior r_Input == D:

                MOVE_RIGHT()

            Elif Warrior _Input = = E:

                ATTACK()

STEP 4 :    If  WIZARD.ATTACK.COLLIDERECT(WARRIOR.RECT):

                    Warrior_Health_bar - = 20

            else

                    CONTINUE


STEP 5 :    return
```

Flowchart

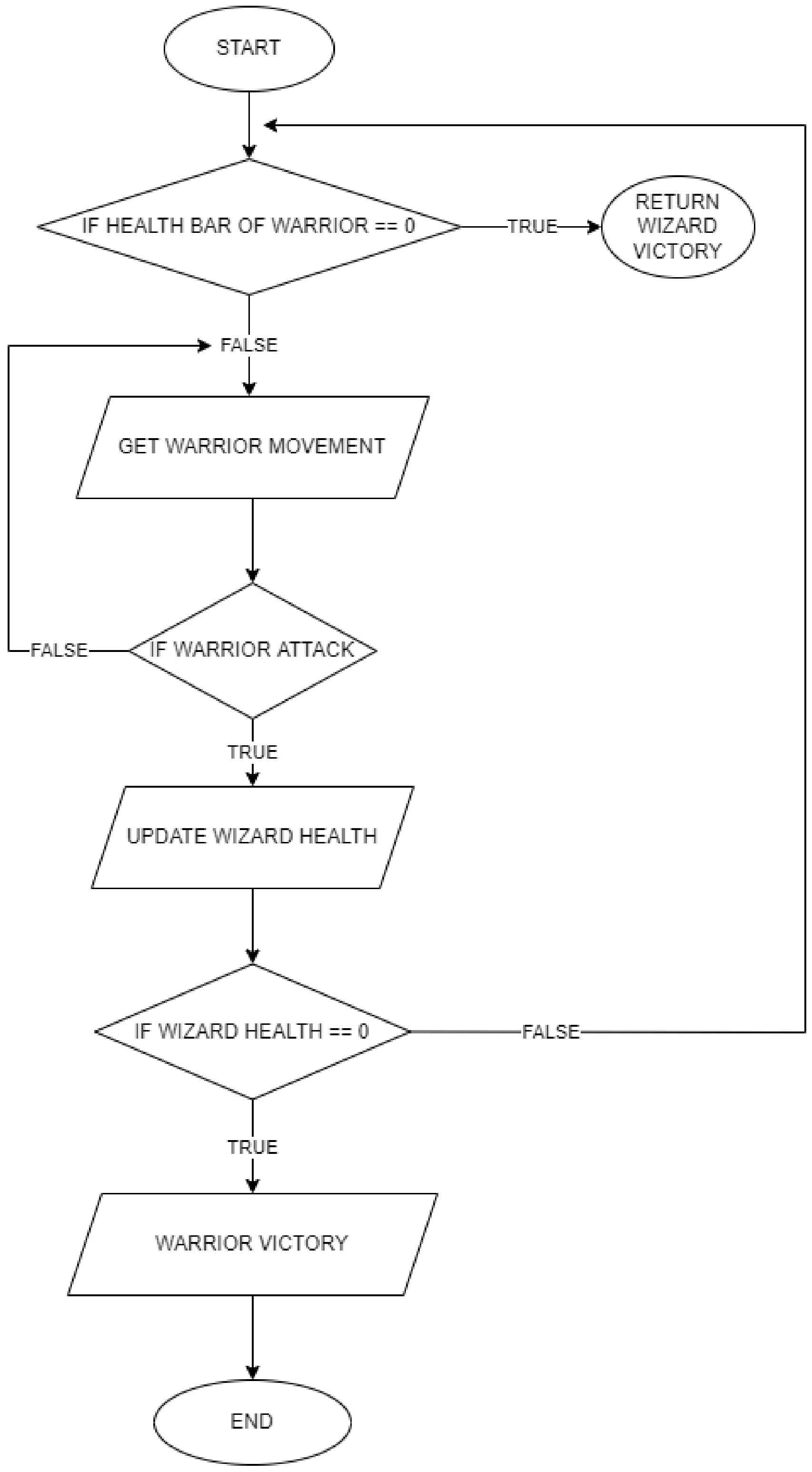

START
IF HEALTH BAR OF WARRIOR == 0
TRUE
RETURN WIZARD VICTORY
FALSE
GET WARRIOR MOVEMENT
IF WARRIOR ATTACK
FALSE
TRUE
UPDATE WIZARD HEALTH
IF WIZARD HEALTH == 0
FALSE
TRUE
WARRIOR VICTORY
END

Fig: Flowchart for Warrior

Pseudocode

```
Warrior ()
While Warrior_Health_bar!= zero:
I=Get_User_Input()
If(i==W)
        JUMP()
Elif(I==A)
        MOVE_RIGHT()
Elif(I==D)
        MOVE_LEFT()
Elif(I==E)
        ATTACK()
If  WIZARD.ATTACK.COLLIDERECT(WARRIOR.RECT):
        Warrior_Health_Bar-=20
Else
        Continue
Return
```

7.3 Algorithm, flowchart & Pseudocode for Wizard :

Algorithm

```
STEP 1 :    Start
STEP 2 :    If health bar of wizard == 0
            return warrior victory
     else
            continue
STEP 3 :    Get_ wizard _Input:
            If  wizard _Input ==UP_ARROW_KEY:
                    JUMP()
            Elif  wizard _Input ==LEFT_ARROW_KEY:
                    MOVE_LEFT()
            Elif  wizard _Input == RIGHT_ARROW_KEY:
                    MOVE_RIGHT()
            Elif wizard _Input==NUM1:
                    ATTACK()
STEP 4 :    If WARRIOR.ATTACK.COLLIDERECT(WIZARD.RECT):
            Wizard_Health_bar -=20
     Else
            Continue
STEP 5 :    Return
```

Flowchart

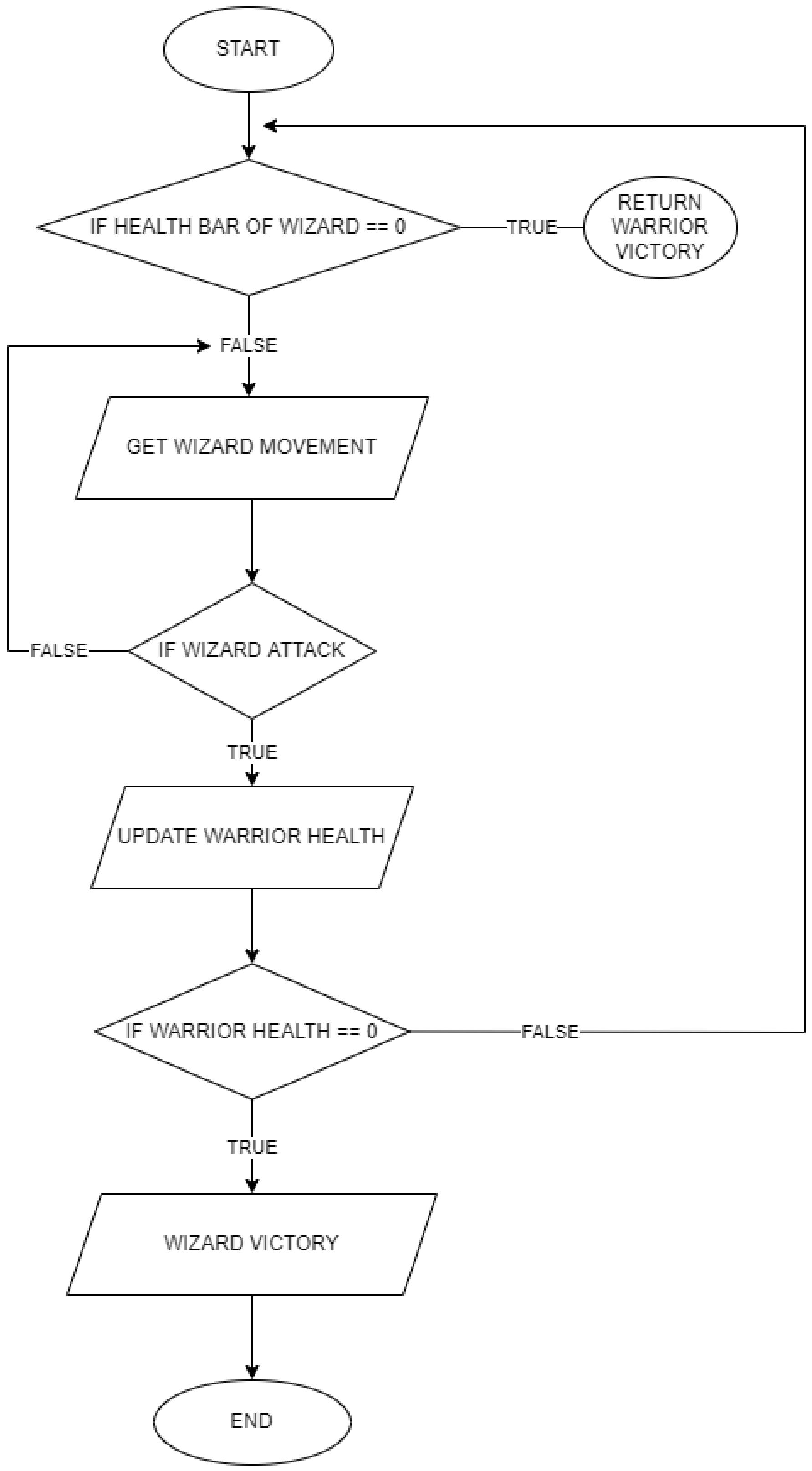


Fig: Flowchart for Wizard.

PseudoCode

```
Wizard ()
While Wizard_Health_bar!= zero:
I=Get_User_Input()
If(i==UP_ARROW_KEY)
    JUMP()
Elif(I==LEFT_ARROW_KEY)
    MOVE_RIGHT()
Elif(I==RIGHT_ARROW_KEY)
    MOVE_LEFT()
Elif(I==NUM1)
    ATTACK()
If  WARRIOR.ATTACK.COLLIDERECT(WIZARD.RECT):
Wizard_Health_Bar-=20
Else
Continue
Return
```

Chapter 8

SOURCE CODE

8.1 game.py

```
from globals import *
from pygame.locals import *
from pygame import mixer
import pygame as pg
from button import Button

# GAME MUSIC

pg.init()
mixer.init()
mixer.music.load('bg_music.mp3')
mixer.music.play(-1)
mixer.music.set_volume(0.15)

window = pg.display.set_mode((1280, 720))

# PyGame initialized
pygame.init()

# Refresh Rate
clock = pygame.time.Clock()

# Font and size
font = pygame.font.Font(font_path, 32)

# For performance
pygame.event.set_allowed([QUIT, KEYDOWN, KEYUP])

# Set window name
pygame.display.set_caption("Stickman")

# Background image
background = pygame.image.load(background_img)
print(background.get_width(), background.get_height())

# Window Dimensions
WINDOW_DIMENSIONS = (background.get_width(), background.get_height())
WINDOW_WIDTH, WINDOW_HEIGHT = WINDOW_DIMENSIONS

# Create screen
flags = DOUBLEBUF
screen = pygame.display.set_mode(WINDOW_DIMENSIONS, flags, 16)
```

```
# Player declaration
player = Stickman_player(WINDOW_DIMENSIONS)


def background_show():
    screen.blit(background, (0, 0))

    # MENU_TEXT = get_font(100).render("MAIN MENU", True, "#b68f40")
    # MENU_RECT = MENU_TEXT.get_rect(center=(640, 100))


def player_show(values):
    player_img = values[0]
    width, height = values[1][0], values[1][1]
    x, y = values[2][0], values[2][1]
    screen.blit(player_img, (x-width/2, y-height))


enemyCharacter = pygame.image.load("EnemyCharacter.png")
enemyX = 900
enemyY = 530

# enemy show function


def enemy_show():
    screen.blit(enemyCharacter, (enemyX, enemyY))


# Player variables
player_img = player.frame_movement()

# Enemy variables
# enemy_img = enemy.frame_movement()

# Moving keys
keys = set([pygame.K_a, pygame.K_LEFT, pygame.K_RIGHT, pygame.K_d])
# keys = set([pygame.K_a, pygame.K_d])

# enemy Moving keys
# keys = set([pygame.K_LEFT, pygame.RIGHT])

# Jump keys
jumpKeys = set([pygame.K_SPACE, pygame.K_w, pygame.K_UP])

# enemy jump keys
# jumpKeys = set([pygame.K_SPACE, pygame.K_w, pygame.K_UP])
```

```
while 1:
    clock.tick(144)

    pressed = pygame.key.get_pressed()

    events = pygame.event.get()
    for event in events:

        # Click on cross button or alt+f4
        if event.type == pygame.QUIT:
            exit()

        # Key down check
        if event.type == pygame.KEYDOWN:
            if event.key == pygame.K_a or event.key == pygame.K_LEFT:
                player.move_left()

            if event.key == pygame.K_d or event.key == pygame.K_RIGHT:
                player.move_right()

            elif event.key == pygame.K_a or event.key == pygame.K_LEFT:
                player.move_left()

            if not player.is_jumping() and event.key in jumpKeys:
                player.jump_up()

        # Key up check
        if event.type == pygame.KEYUP:
            if event.key in keys:
                player.stop_moving()
            if event.key == pygame.K_l:
                print(player.get_pos())

    # Get player image
    player_img = player.frame_movement()

    # Final display of all images
    screen.fill((0, 0, 0))
    background_show()
    enemy_show()
    player_show(player_img)

    pygame.display.update()


def get_font(size): # Returns Press-Start-2P in the desired size
    return pygame.font.Font("font.ttf", size)
```

```
def back():
    while True:
        PLAY_MOUSE_POS = pygame.mouse.get_pos()
        PLAY_BACK = Button(image=None, pos=(500, 50), text_input="BACK",
font=get_font(20), base_color="White",
                           hovering_color="Green")
        PLAY_BACK.changeColor(PLAY_MOUSE_POS)
        PLAY_BACK.update(screen)

        for event in pygame.event.get():
            if event.type == pygame.QUIT:
                pygame.quit()
                sys.exit()
            if event.type == pygame.MOUSEBUTTONDOWN:
                if PLAY_BACK.checkForInput(PLAY_MOUSE_POS):
                    mixer.music.stop()
                    import main
        pygame.display.update()


print("over")
```

8.2 globals.py

```
import sys
import os
import random
import time
import copy

sys.path.append("src")
sys.path.append("img")
sys.path.append("music")
sys.path.append("fonts")

from Player import *
```

8.3 main.py

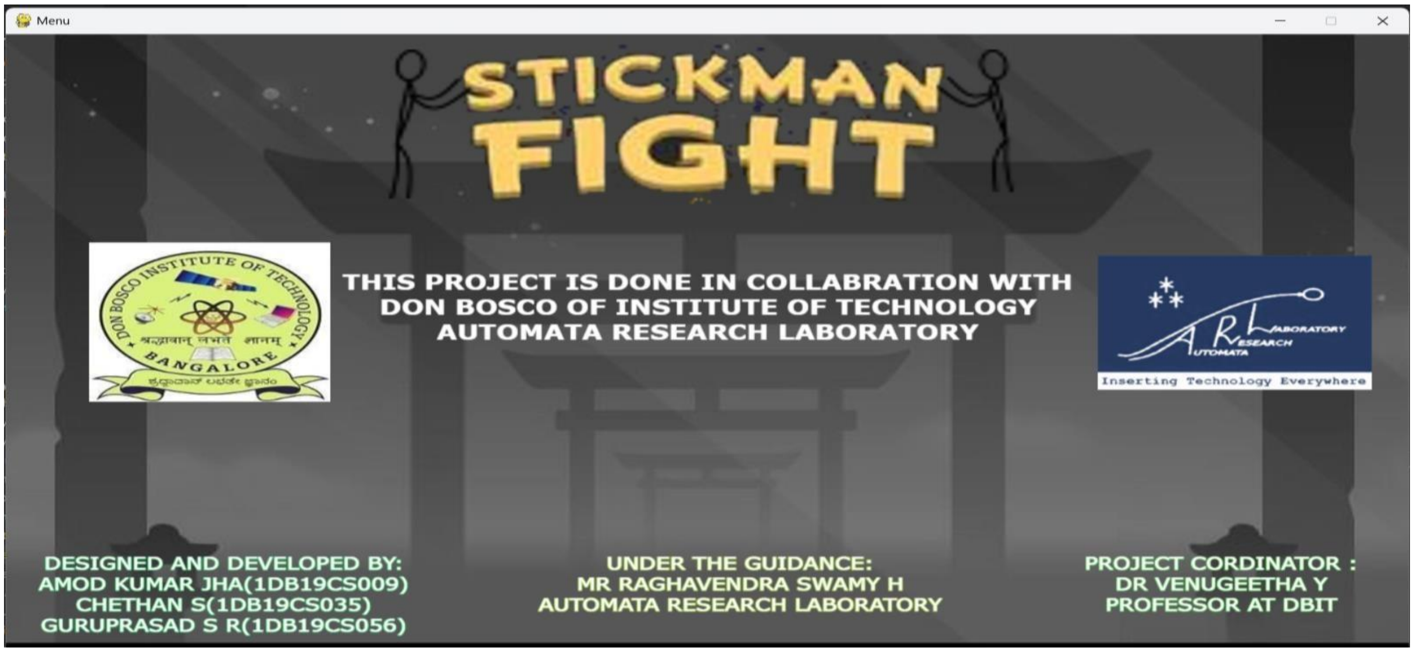
```
from traceback import format_tb
import pygame, sys
from button import Button
from globals import *
from pygame.locals import *
from pygame import mixer
```

```python
# MENU MUSIC

pygame.init()
mixer.init()
mixer.music.load('MenuMusic.mp3')
mixer.music.play(-1)
mixer.music.set_volume(0.50)

SCREEN = pygame.display.set_mode((1280, 720))
pygame.display.set_caption("Menu")
TITLE =  pygame.image.load("title.png")
BG = pygame.image.load("assets/Menu.png")
CR=pygame.image.load("assets/Credits.jpeg")
LOGO=pygame.image.load("assets/dbit_logo.jpg")
MIT=pygame.image.load("assets/MIT License.jpeg")
MainTitle=pygame.image.load("assets/MainTitle.png")




def get_font(size): # Returns Press-Start-2P in the desired size
    return pygame.font.Font("font.ttf", size)


# TO BE USED - CREDITS FONT
def credit_font(size):
    return pygame.font.Font("times new roman bold.ttf",size)


def play():
    while True:
        import game

def options():
    while True:
        OPTIONS_MOUSE_POS = pygame.mouse.get_pos()

        SCREEN.fill("whitesmoke")
        HELP_TEXT = get_font(100).render("HOW TO PLAY", True, "Black")
        HELP_RECT = HELP_TEXT.get_rect(center=(640, 50))

        HELP_TEXT1 = get_font(60).render("To Move Forward and Backward ,Use W-A-S-D
or Arrow keys", True, "Black")
        HELP_RECT1 = HELP_TEXT1.get_rect(center=(640, 150))
        HELP_TEXT2 = get_font(60).render("To Jump ,use either SPACEBAR or W or UP-
Arrow", True, "Black")
        HELP_RECT2 = HELP_TEXT2.get_rect(center=(640, 250))
        HELP_TEXT3 = get_font(60).render("To fight Use the Left Mouse Button ", True,
"Black")
        HELP_RECT3 = HELP_TEXT3.get_rect(center=(640, 350))
        HELP_TEXT4 = get_font(60).render("You Will have three lives.Be Cautious about
the health", True, "Black")
        HELP_RECT4 = HELP_TEXT4.get_rect(center=(640, 450))
        SCREEN.blit(HELP_TEXT, HELP_RECT)
        SCREEN.blit(HELP_TEXT1, HELP_RECT1)
        SCREEN.blit(HELP_TEXT2, HELP_RECT2)
        SCREEN.blit(HELP_TEXT3, HELP_RECT3)
        SCREEN.blit(HELP_TEXT4, HELP_RECT4)

        OPTIONS_BACK = Button(image=None, pos=(640, 600), text_input="BACK",
font=get_font(75), base_color="Black",
                              hovering_color="Green")

        OPTIONS_BACK.changeColor(OPTIONS_MOUSE_POS)
        OPTIONS_BACK.update(SCREEN)

        for event in pygame.event.get():
            if event.type == pygame.QUIT:
                pygame.quit()
                sys.exit()
            if event.type == pygame.MOUSEBUTTONDOWN:
                if OPTIONS_BACK.checkForInput(OPTIONS_MOUSE_POS):
                    main_menu()

        pygame.display.update()


def credits():
    while True:

        SCREEN.blit(CR, (0, 0))

        OPTIONS_MOUSE_POS = pygame.mouse.get_pos()
        OPTIONS_BACK = Button(image=None, pos=(1100, 650), text_input="BACK",
font=get_font(60), base_color="Black",
                              hovering_color="Green")
        OPTIONS_BACK.changeColor(OPTIONS_MOUSE_POS)
        OPTIONS_BACK.update(SCREEN)

        for event in pygame.event.get():
            if event.type == pygame.QUIT:
                pygame.quit()
                sys.exit()
            if event.type == pygame.MOUSEBUTTONDOWN:
                if OPTIONS_BACK.checkForInput(OPTIONS_MOUSE_POS):
                        main_menu()

                pygame.display.update()



SCREEN.blit(MainTitle,(0,0)) # render MIT license to screen
pygame.display.flip()
pygame.event.pump()
pygame.time.delay(3000) # continue to show screen for 3 seconds



SCREEN.blit(MIT,(0,0)) # render MIT license to screen
pygame.display.flip()
pygame.event.pump()
pygame.time.delay(3000) # continue to show screen for 3 seconds


def main_menu():
    while True:

        SCREEN.blit(BG, (0, 0))
        SCREEN.blit(TITLE, (0, 0))


        MENU_MOUSE_POS = pygame.mouse.get_pos()

        # MENU_TEXT = get_font(100).render("MAIN MENU", True, "#b68f40")
        # MENU_RECT = MENU_TEXT.get_rect(center=(640, 100))

        PLAY_BUTTON = Button(image=pygame.image.load("assets/Play Rect.png"),
pos=(640, 350),
                            text_input="PLAY", font=get_font(75),
base_color="#d7fcd4", hovering_color="White")
        OPTIONS_BUTTON = Button(image=pygame.image.load("assets/Options Rect.png"),
pos=(640, 500),
                            text_input="OPTIONS", font=get_font(75),
base_color="#d7fcd4", hovering_color="White")
        QUIT_BUTTON = Button(image=pygame.image.load("assets/Quit Rect.png"),
pos=(640, 650),
                            text_input="QUIT", font=get_font(75),
base_color="#d7fcd4", hovering_color="White")
        CREDITS_BUTTON= Button(image=pygame.image.load("assets/Credits
Rect.png"),pos=(1100,650),text_input="CREDITS",font=get_font(50),base_color="#d7fcd4"
,hovering_color="White")

        # SCREEN.blit(MENU_TEXT, MENU_RECT)
        # SCREEN.blit(MENU_RECT)

        for button in [PLAY_BUTTON, OPTIONS_BUTTON, QUIT_BUTTON,CREDITS_BUTTON]:
            button.changeColor(MENU_MOUSE_POS)
            button.update(SCREEN)

        for event in pygame.event.get():
            if event.type == pygame.QUIT:
                pygame.quit()
                sys.exit()
            if event.type == pygame.MOUSEBUTTONDOWN:
                if PLAY_BUTTON.checkForInput(MENU_MOUSE_POS):
                    play()
                if OPTIONS_BUTTON.checkForInput(MENU_MOUSE_POS):
                    options()
                if CREDITS_BUTTON.checkForInput(MENU_MOUSE_POS):
                    credits()
                if QUIT_BUTTON.checkForInput(MENU_MOUSE_POS):
                    pygame.quit()
                    sys.exit()

        pygame.display.update()

main_menu()
```

Chapter 9

SCREENSHOTS

9.1 Home


Menu
STICKMAN FIGHT
DON BOSCO INSTITUTE OF TECHNOLOGY
BANGALORE
Inserting Technology Everywhere

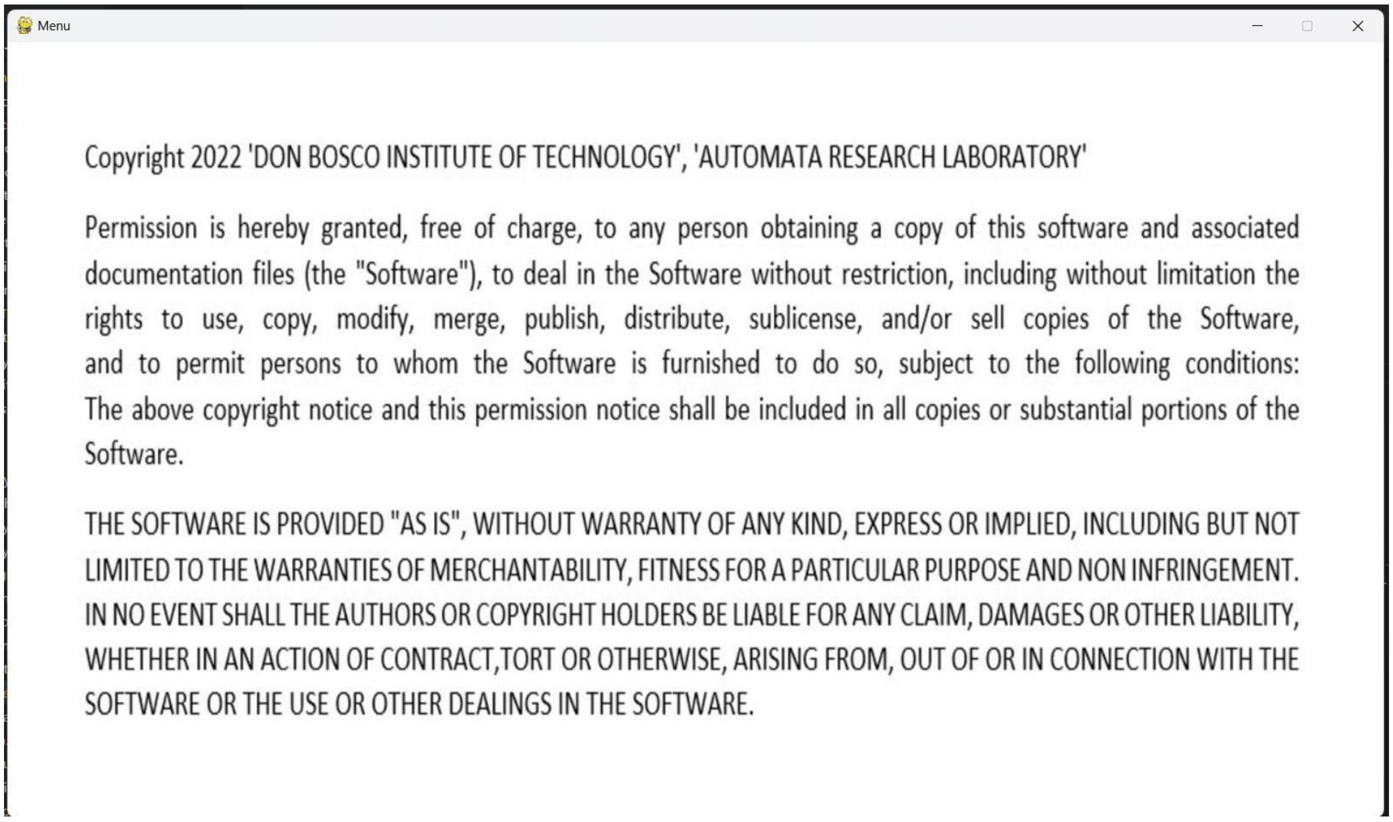
THIS PROJECT IS DONE IN COLLABRATION WITH
DON BOSCO OF INSTITUTE OF TECHNOLOGY
AUTOMATA RESEARCH LABORATORY

DESIGNED AND DEVELOPED BY:
AMOD KUMAR JHA(1DB19CS009)
CHETHAN S(1DB19CS035)
GURUPRASAD S R(1DB19CS056)

UNDER THE GUIDANCE:
MR RAGHAVENDRA SWAMY H
AUTOMATA RESEARCH LABORATORY

PROJECT CORDINATOR :
DR VENUGEETHA Y
PROFESSOR AT DBIT

9.2 MIT License

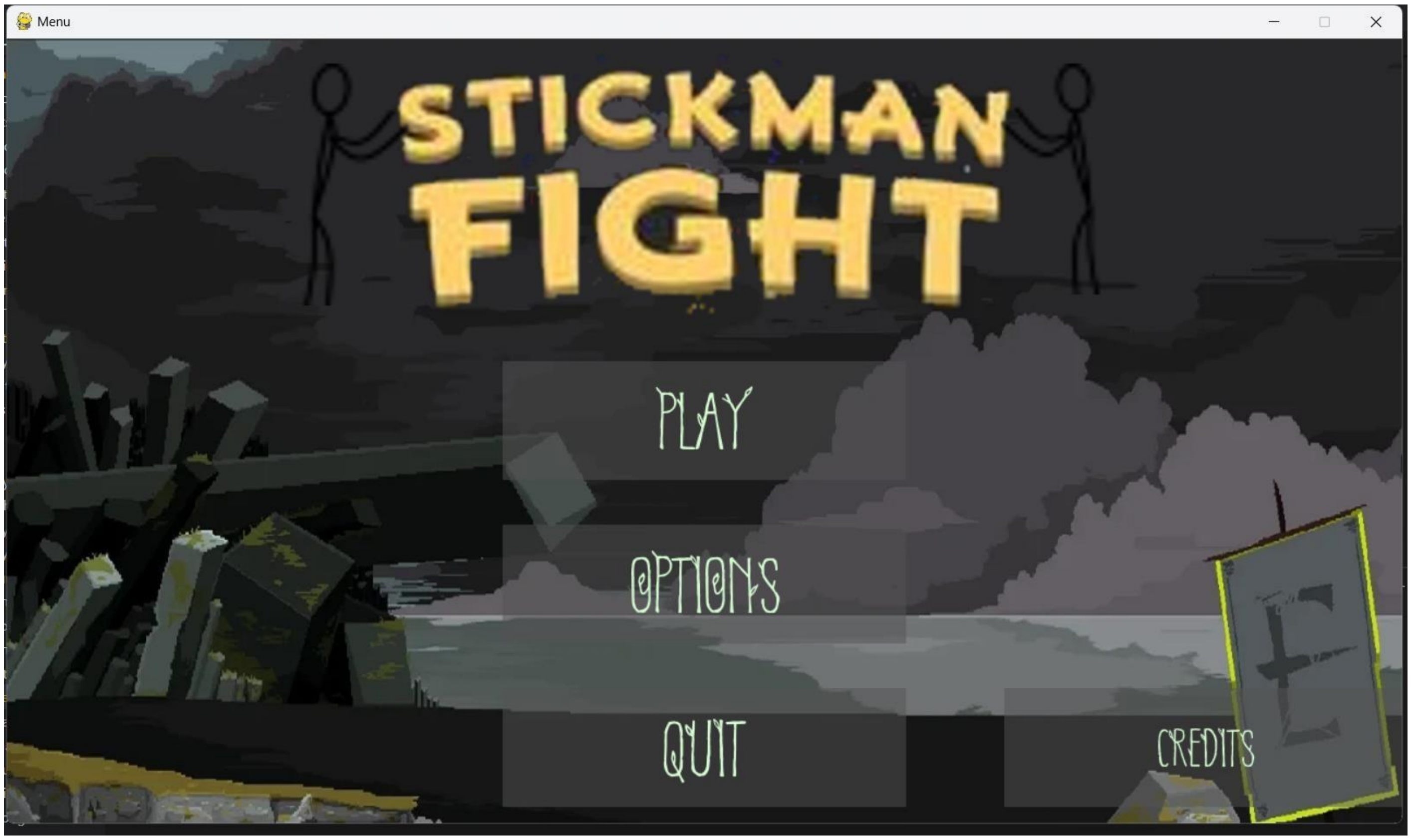


Copyright 2022 'DON BOSCO INSTITUTE OF TECHNOLOGY', 'AUTOMATA RESEARCH LABORATORY'

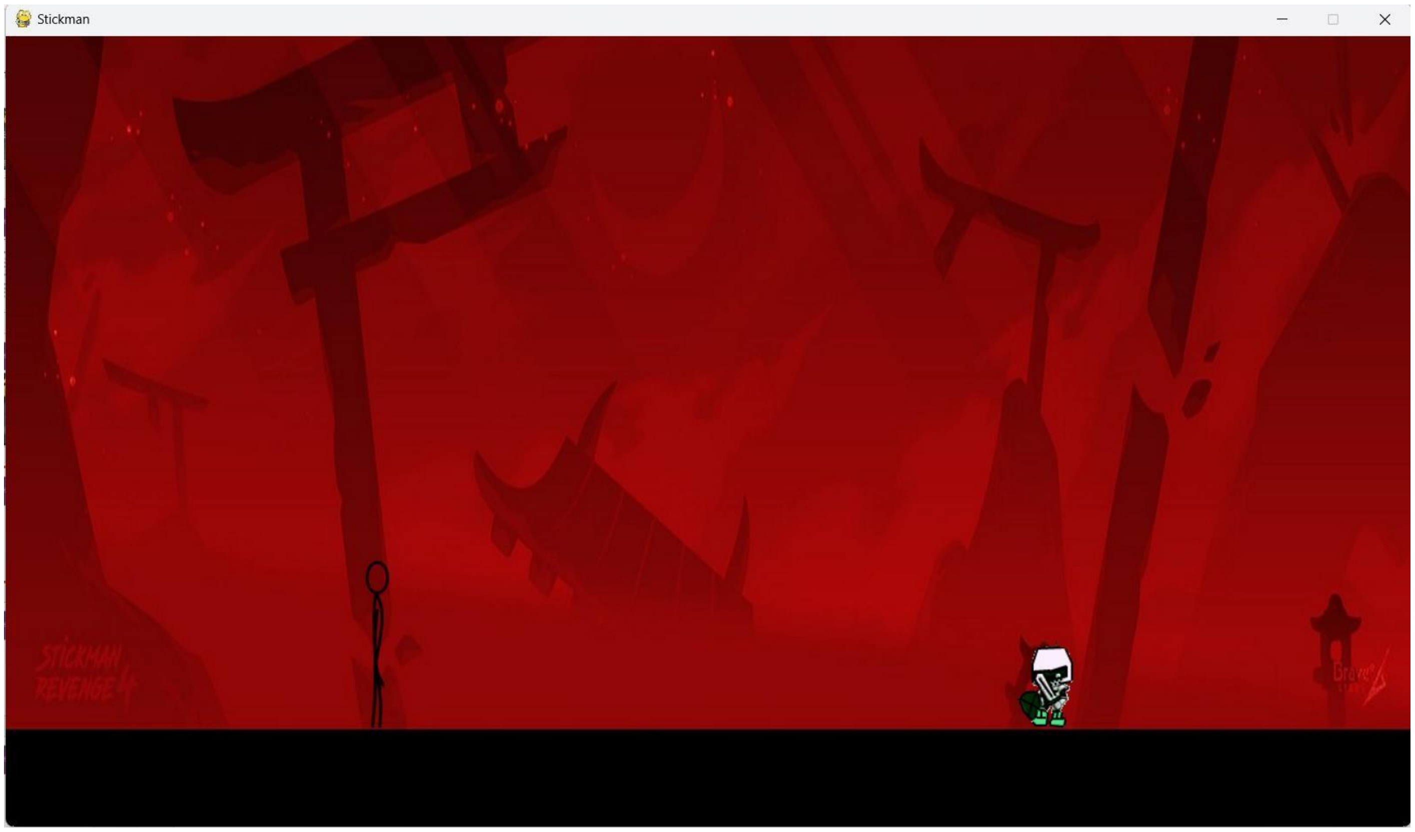
Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions: The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NON INFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

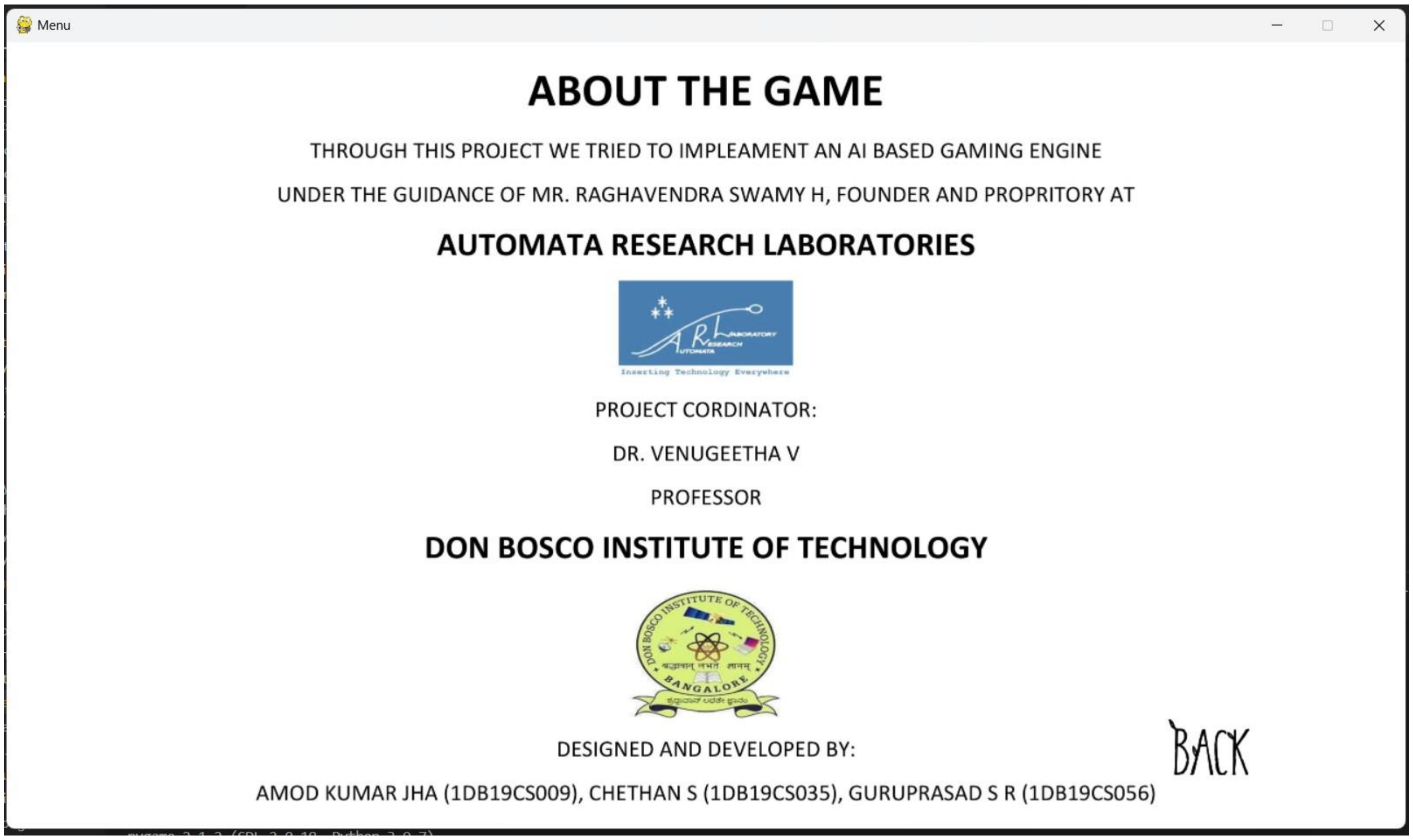
9.3 Home Window



9.4 Stickman Window



9.5 Credit



Menu

ABOUT THE GAME

THROUGH THIS PROJECT WE TRIED TO IMPLEAMENT AN AI BASED GAMING ENGINE

UNDER THE GUIDANCE OF MR. RAGHAVENDRA SWAMY H, FOUNDER AND PROPRITORY AT

AUTOMATA RESEARCH LABORATORIES

PROJECT CORDINATOR:

DR. VENUGEETHA V

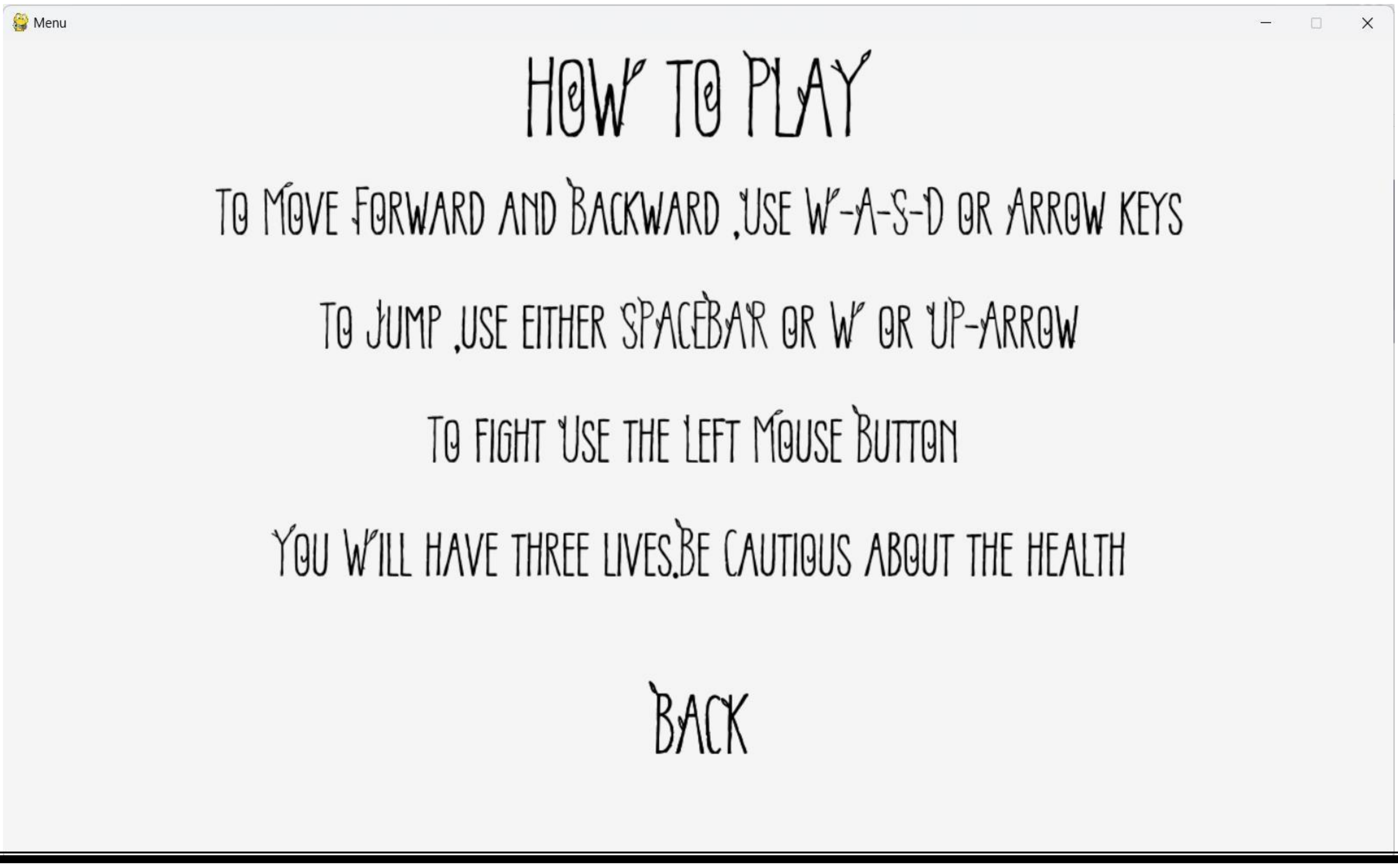
PROFESSOR

DON BOSCO INSTITUTE OF TECHNOLOGY

DESIGNED AND DEVELOPED BY:

AMOD KUMAR JHA (1DB19CS009), CHETHAN S (1DB19CS035), GURUPRASAD S R (1DB19CS056)

9.6 Options



Menu

HOW TO PLAY

TO MOVE FORWARD AND BACKWARD ,USE W-A-S-D OR ARROW KEYS

TO JUMP ,USE EITHER SPACEBAR OR W OR UP-ARROW

TO FIGHT USE THE LEFT MOUSE BUTTON

YOU WILL HAVE THREE LIVES.BE CAUTIOUS ABOUT THE HEALTH

Chapter 10

INTERNSHIP ACTIVITY

- The internship activity was about watching the movie "**The Intern**", which was released on **25th, September 2015**.
- This movie is about a seventy-year-old widower Ben Whittaker who had discovered that retirement wasn't all it's cracked up to be. Seizing an opportunity to get back in the game, he becomes a senior intern at an online fashion site, founded and run by Jules Ostin.
- The movie explains themes of generational wisdom, not judging people by their age (or anything superficial), and making the most of people's talents.
- It is also a movie about starting over and doing what we enjoy.
- It explains the importance of interns in a company and not making decisions that are not satisfactory to one's self.

Chapter 11

Key Take Aways

- Learnt about Project development life cycle and all the components that go into developing a full fledged product.
- Learnt about team management, and everything that it takes to make a successful team project.
- Learnt about core concepts in game development like pygame and other python supported libraries.
- Learnt about the significance of Artificial Intelligence in the gaming industry
- Learnt about effective communication among the team members, which is significant in bringing out and implementing ideas related to project.

REFERENCES

- **Installation of Python IDE - https://www.python.org/downloads/**
- **Installation of PyCharm - https://www.jetbrains.com/pycharm/download/**
- **Pygame package - https://www.pygame.org/**
- **https://analyticsindiamag.com/top-9-python-frameworks-for-game-development/**
- **https://pygame-ai.readthedocs.io/en/latest/index.html**
- **YouTube - https://youtu.be/s5bd9KMSSW4**
- **https://opensource.org/licenses/MIT**
- **https://www.synopsys.com/blogs/software-security/5-types-of-software-licenses-you-need-to-understand/**