Roboviz

By

Salahuddin
Ahmad Javaid
Zohair Adnan
Ibrahim Khan

Submitted to Faculty of Department of Computer Software Engineering National University of Sciences and Technology, Islamabad in partial fulfilment for the requirements of a B.E Degree in Computer Software Engineering, June 2018

In the name of Allah, the Most Beneficent, the Most Merciful

ABSTRACT

This project is intended to provide a deep neural network model using a variety of machine learning techniques, including computer vision, reinforcement tree learning, and case-based reasoning. We are implementing these techniques to form a novel approach for training AI-bot on its gaming environment. Bot will use computational methods to "learn" information directly from data without relying on a predetermined equation as a model. The algorithms adaptively improve their performance as the number of samples available for learning increases.

The product is a project that will use "Keras library" of neural network model to train model using different reinforcement techniques. Objects will be recognized through computer vision. Game will be developed using gaming development toolkit. Training API's will train the model by using GPU computational power. Bot will be added to the game.

CERTIFICATE FOR CORRECTNESS AND APPROVAL

It is certified that work contained in the thesis Roboviz carried out by Salahuddin, Ibrahim Khan, Ahmad Javaid and Zohair Adnan under supervision of Dr Naima for partial fulfilment of Degree of Bachelor of Software Engineering is correct and approved.

Approved By

Dr Naima

Department of CSE, MCS

Dated:

_____

DECLARATION

No portion of the work presented in this dissertation has been submitted in support of another award or qualification either at this institution or elsewhere.

DEDICATION

To our parents, without whose support and cooperation, a work of this magnitude would not have been possible. To our supervisor, Dr Naima who has given us great support and valuable suggestions throughout the implementation process.

ACKNOWLEDGEMENTS

There is no success without the will of ALLAH Almighty. We are grateful to ALLAH, who has given us guidance, strength and enabled us to accomplish this task. Whatever we have achieved, we owe it to Him, in totality. We are also grateful to our parents and family and well-wishers for their admirable support and their critical reviews. We would like to thank our supervisor. Dr Naima, for her continuous guidance and motivation throughout the course of our project. Without their help, we would have not been able to accomplish anything.

Table of Contents

List of Figures

List of Tables

Chapter 1 Introduction

1.1. Overview

This project aims to provide a deep neural network model using a number of machine learning techniques, including computer vision, deep learning, reinforcement tree learning, and case-based reasoning. We are implementing these techniques to form a new or novel approach for training AI-bot on its gaming environment. Bot will use computational techniques to "learn" information directly from data(interaction) without depending on a model in the form of predetermined equation. The algorithms surely and certainly improve their performance and accuracy as the variety of samples available increases.
The product is a project that will use "Keras library" of neural network model to train model using different reinforcement techniques. Objects will be recognized through computer vision. Game will be developed using gaming development toolkit. Training API's will train the model by using GPU computational power. Bot will be added to the game..

1.2. Problem Statement

Many of the practical things around us need persistent practice before it can be perfected.
With practice comes perfection, but each trial and error has its own cost and takes time. Sometimes that cost is too much and cannot be paid or is not feasible to be paid. There are cases where something needs to be learned quickly but cannot be done without all the practice. All can be solved using a simulated environment.

1.3. Approach

We have proposed simulated learning through a gaming environment. Our learning is purely interaction based. Knowledge about the game is not fed to the bot. The bot is thrown in the game and learns how to play through trial and error. The player playing against the bot is giving the information the bot needs to learn the game By the use of a simulated environment we are saving cost as well as time that could have been spent on the learning process.

1.4. Scope

At the moment we are dealing only with 2d games only. There are a number of game which the bot will be able to play. So there is not one specific game. The bot will play against the payer. At the start the bot will have very little knowledge about the game. As the player will play the bot will continue to learn and improve with each game. The bot will be rewarded points whenever a good game is played and vice versa. In this way we are achieving interaction based learning in a gaming environment.

1.5. Objectives

This project is intended to provide a deep neural network model using a number of machine learning techniques, including computer vision, deep learning, decision tree learning, and case-based reasoning.

During the course of this project, all the aspects of software engineering are covered i.e. survey and feasibility analysis, requirement gathering, architectural and detailed design, implementation and testing along with documentation (SRS, SDS, Test Document, Final Report and User manual). Students are also expected to develop extensive knowledge and technical skills in the following fields:

1. Machine learning.
2. Deep neural network modeling.
3. Decision tree learning.
4. Open source software understanding.
5. Computer vision.
6. AI based learning.

1.6. Deliverables

Sr.	Tasks	Deliverables
1	**Literature Review**	**Literature Survey**
2	**Requirements Specification**	**Software Requirements Specification document (SRS)**
3	**Detailed Design**	**Software Design Specification document (SDS)**
4	**Implementation**	**Project demonstration**
5	**Testing**	**Evaluation plan and test document**

6	**Training**	**Deployment plan**

7	**Deployment**	**Complete application with necessary documentation**

Table 1-1 Deliverables

1.7. Overview of the Document

This document shows the complete working process of our application Roboviz. It starts off with the literature review which shows past work done in similar field, requirement analysis of the system, system architecture which highlights the modules of the software and represents the system in the form of component diagram, Use Case Diagram, Sequence Diagram and general design of the system. Then it will move on to discuss the detailed Description of all the components involved. Further the dependencies of the system and its relationship with other products and the capacity of it to be reused will be discussed. At the end test cases and any future work proposal has been presented.

1.8. Document Conventions

Heading are prioritized in a numbered fashion, the highest priority heading having a single digit and subsequent headings having more numbers, according to their level. Font used is Times New Roman. All the main headings are of size 18 and bold. All the second level sub-headings are of size 16 and bold. All the further sub-headings are of size 14 and bold. All references in this document are provided where necessary, however where not present, the meaning is self-explanatory. All ambiguous terms have been clarified in the glossary at the end of this document.

1.9. Intended Audience

The audience for this document is:

1. **Developers:** (Project Group)

   In order to be certain that they are developing the right project that fulfills the requirements provided in this document.

2. **Testers:** (Project Group, Supervisor)

   To have an exact list of the features and functions that must respond according to requirements.

3. **Users:**

So to get familiar with the idea of the project and how to use/respond in failure situations and improve other features that would make it even more efficient.

4. **Documentation writers:** (Project Group)

   To know what features and in what way they have to explain. What technologies are required, how the system will respond in each user's action, what possible system failures may happen and what are the solutions to all those failures etc.

5. **Project Supervisor:** (Dr Naima)

   This document will be used by the project supervisor to check whether all the requirements have been understood and in the end whether the requirements have been implemented properly and completely.

6. **Project Evaluators:** (CSE Dept. MCS)

   So to know the scope of the project and evaluate the project throughout the development for grading.

Chapter 2 Literature Review

Following are a few of the works that have been done in the field of artificial intelligence and reinforcement learning.
For human beings, chess may additionally take an entire life to grasp. But Google DeepMind's new synthetic intelligence software, AlphaZero, can teach itself to triumph over the board in a count number of hours.

Building on its past success with the AlphaGo suite—a chain of laptop packages designed to play the Chinese board sport Go—Google boasts that its new AlphaZero achieves a stage of "superhuman performance" at now not simply one board recreation, however 3: Go, chess, and shogi (essentially, Japanese chess). The group of pc scientists and engineers, led with the aid of Google's David Silver, mentioned its findings lately in the magazine Science.

"Before this, with system getting to know, you could get a system to do precisely what you need—but only that component," says Ayanna Howard, an professional in interactive computing and artificial intelligence at the Georgia Institute of Technology who did not take part within the studies. "But AlphaZero indicates that you can have an algorithm that isn't so [specific], and it could examine within positive parameters.

AlphaZero's smart programming in reality united states of americathe ante on gameplay for human and gadget alike, however Google has lengthy had its sights set on some thing bigger: engineering intelligence.

The researchers are cautious not to claim that AlphaZero is on the verge of global domination (others were a bit faster to leap the gun). Still, Silver and the rest of the DeepMind squad are already hopeful that they'll at some point see a similar machine applied to drug layout or materials technology.

Gameplay has long been revered as a gold widespread in synthetic intelligence studies. Structured, interactive games are simplifications of actual-world eventualities: Difficult choices must be made; wins and losses force up the stakes; and prediction, crucial questioning, and approach are key.

Encoding this sort of ability is difficult. Older sport-gambling AIs—consisting of the first prototypes of the authentic AlphaGo—have traditionally been pumped complete of codes and records to mimic the revel in usually earned thru years of herbal, human gameplay (essentially, a passive, programmer-derived expertise sell off). With AlphaGo Zero (the maximum latest version of AlphaGo), and now AlphaZero, the researchers gave this system just one input: the guidelines of the game in query. Then, the machine hunkered down and actively discovered the tricks of the change itself.

Chapter 3 Software Req. Specification (SRS)

3.1. Introduction

The introduction of the Software Requirements Specification (SRS) aims to show or provides an overview of the complete SRS with purpose, scope, aim, definitions, acronyms, abbreviations and references. The purpose of this document is to present detailed and complete description of the Roboviz by explaining the problem statement in detail. The elaborate requirements of the Roboviz are provided in this document.

3.1.1. Purpose

This document includes software requirements for Roboviz. The introduction of the Software Requirements Specification (SRS) provides an overview of the entire SRS with purpose, scope, definitions, acronyms, abbreviations, references and overview of the SRS. The aim of this document is to present detailed description of the project Roboviz in which the system will provide a bot that is a type of AI expert system software that plays a video game in the place of a human .Bot will be thrown into a virtual environment where it teaches itself through machine learning known as reinforcement learning, how to achieve their goals through trial and error.

3.1.2. Project Vision

Many of the practical things around us need persistent practice before it can be perfected.
With practice comes perfection, but each trial and error has its own cost and takes time. Sometimes that cost is too much and cannot be paid or is not feasible to be paid. There are cases where something needs to be learned quickly but cannot be done without all the practice. All can be solved using a simulated environment.

For	**Various category of players**
What	**Gaming bot.**

The	**Roboviz**
Is	**Desktop application**
That	**Provides training in a simulated environment.**

Table 3-1 Project Vision

3.2. Overall Description

3.2.1. Product Perspective

Roboviz will focus on the interaction based learning. The main aim is to enable a bot to learn from the environment instead of being fed the information. Through this real life situations which require a lot of practice and time for learning can be easily handled in a simulated environment.

3.2.2. Product Features:

Main features of the application are listed below.

1. Selection of game (environment).
2. Generating or creating a photorealistic or non photorealistic image(environment) from a 2D model.
3. Interpreting frames.
4. Extraction of useful information from the environment.
5. Getting inputs from the human player and translating them into useful and meaningful actions in the simulated environment.
6. Learnig from the interaction in the environment.

3.2.3. User Classes and Characteristics

Following are user classes and their brief description.

Game

This class is responsible for the selection of the game. That is the selection of the learning environment. The other purpose of this is rendering. Rendering or image synthesis is the automatic process of generating a photorealistic or non-photorealistic image from a 2D or 3D model (or models in what collectively could be called a scene file) by means of computer programs. Also, the results of displaying such a model can be called a render. A scene file contains objects in a strictly defined language or data structure; it would contain geometry, viewpoint, texture, lighting, and shading information as a description of the virtual scene. The data contained in the scene file is then passed to a rendering program to be processed and output to a digital image or raster graphics image file.

Screen Capture

This class is responsible for all the functions related to frames per second of the game. They will include the functionalities like get Frame and get frame difference.

Visual preprocessor

This class is responsible for the application of different filters on the images being rendered. This is used to extract features from images. A convolution is an operation that changes a function into something else. We do convolutions so that we can transform the original function into a form to get more information. Convolutions have been used for a long time in image processing to blur and sharpen images, and perform other operations, such as, enhance edges and emboss.

Input injector

This class contains functions related to getting inputs from the human player through keyboard or mouse and translating them into meaningful actions in the simulated environment.

3.2.4. Operating environment

A single cuda compatible computer environment and a 2D game is required.

Hardware

RAD operates, either directly or indirectly, with the following external hardware:

1. GPU capable custom made PC.
2. More than 4 GPU memory.

3. Internet connection.

Software

1. Linux operating system.
2. Single algorithmic scaling to multiple games

3.2.5. Design and Implementation Constraints

The design and implementation constraints are as follows

1. Hardware is constrained to a single GPU.
2. The GPU is 1080Ti.
3. Software wise we are limited to 2D games only.
4. It is not able to entertain multiple users at a time.

3.2.6. User Documentation

A user manual will be provided to the users in which separate instructions will be given according to the particular user i.e., player, developers and testers. It will include the details of the software working. Help documents will also be a part of the system. The project report will also be available for the users which will highlight the software's features, working and procedures.

3.2.7. Assumptions and Dependencies

1. User owns a compatible hardware like a GPU compatible personal computer.
2. User should know how to use basic gaming applications.
3. The customer knows the language (English) used in the user interface to perform actions.

3.3. External Interfaces Requirements

3.3.1. User Interfaces

1. User interface will be displayed on the personal computer.
2. Main menu for navigation will be used.
3. Interface will be user friendly and the standard English-US will be used

3.3.2. Hardware Interfaces

1. GPU compatible PC.
2. More than 4 GPU memory required.

3.3.3. Software Interfaces

1. The application will run on linux operating system.

2. Single algorithmic scaling to multiple games is required.

Data Input

1. Action selection from the given choices by the user in respective games.

3.3.4. Communications Interfaces

Roboviz will not require communication interface as it is a standalone application.

3.4. System Features

This section explains in detail the system features of the Roboviz.

3.4.1. Accessing the Main Menu

Description and priority

After starting the application the main menu will be displayed.

Its priority will be high as without this feature the application will not be navigable and the user will not be able to perform any action.

Stimulus/Response Sequence

1. Open the application.

2. Access the main menu.

Functional requirements

REQ-1: Application shall be properly installed on the PC.

REQ-2: The different options available shall be

1. Train
2. Play game
3. Help menu
4. Exit

REQ-3: At any time user shall be able to exit the application when required.

3.4.2 Game play

Description and priority

This holds a very high priority since it is a core feature of this product.

Stimulus/Response Sequence

1. The user selects game option from main menu.
2. User selects one of the actions using controllers.
3. User plays the game and proceeds to the next.
4. System then takes user to view result screen after the completion of the game.

Functional requirements

REQ-4: List of the games shall be displayed for user to select.

REQ-5: User shall select the game to be played.

REQ-6: Set of instructions shall be provided on how to play the game after selecting a particular game.

REQ-7: User shall be able to replay a particular game.

REQ-8: The score of the game will be displayed to the player at the end of each game.

REQ-9: User shall be able to exit the game and move to main menu when required.

3.4.3 Help Menu

Description and Priority

Help menu has a medium or low priority. It will have all the instructions and rules required to use the application.

Response Sequences

1. The user chooses Help Menu option from Main Menu.
2. An instruction manual is opened to guide the user.

Functional Requirements

REQ-11: Instruction Manual will be shown on choosing the Help option.

3.5. Other Nonfunctional Requirements

3.5.1. Resolution low

The resolution of the game is kept low as high resolution games require a higher processing power. Thus it is more feasible to keep the resolution low in order to ensure good performance.

3.5.2. Lag free operation

It is very important that the game is lag free. It will provide the user a suitable environment for playing the game. If the game starts to lag the performance is affected. There is no point of playing a game that lags. Incase the game lags it will become difficult to play the game.

3.5.3. Performance Requirement

The response time of Roboviz should be less. It should be rapid and fast. The player should not have to wait after giving a command. Recovery time of the application should be less in case it crashes and a restart is required.

3.5.4. Software Quality Attributes

Usability

The graphical user interface of app is to be designed with usability as the first priority. The app will be presented and organized in a manner that is both visually appealing and easy for the user to navigate keeping in view the variety of players.

Reliability

Application shall provide reliability to the user. The product will run stably with all the features mentioned above available and executing perfectly. It shall be tested and debugged completely. All exceptions shall be well handled.

Portability

In API, portability can be defined as “compatibility of application with platform upgraded or downgraded versions. In linux platform when an up gradation is done the application will require some changes for compatibility with the new version..

Flexibility

The design and architecture of the application will be flexible enough for catering any new requirements, if any at some later stage or for the application enhancement.

Scalability

The application is expected to handle one user at a time. One instance of the application could be opened on a phone at a time.

Availability

The application will be available 24/7, provided mobile is in working state and the application is installed and configured properly.

Chapter 4 Design and Development

4.1. Introduction

This design document covers all our functional requirements and demonstrates how they interrelate with each other abstractly. The low level design also illustrates as to how we have been implementing and how we are going to implement all of these requirements. This low level design for the time being does not address any non-functional requirements that our system has and that have been mentioned in the SRS Document.

4.2. Purpose

The purpose or aim of this Software Design Document is to deliver a portrayal of the design of our system adaquate enough to allow or enable the software development to start with an understanding of what is to be made and how it is supposed to be developed. This Software Design Document shows information essential to get a description of the minute details for the software and the system to be built. The purpose of this document is to present a design view and detailed description of RoboViz. It will explain the purpose, features, interfaces, what the system will do, its entire processes in detail, the constraints or limitations under which it should operate and how the system will react to inputs and what will be its outputs. This document is a guide for both the stakeholders and the system developers.

4.3. Project Scope

This project aims to provide a deep neural network model using a variety of machine learning techniques, including computer vision, deep learning, decision tree learning, and case-based reasoning. We are implementing these techniques to form a unique approach for training AI-bot on its gaming environment. Bot will use computational ways to “learn” information directly from data without depending model in form of a predetermined equation. The algorithms surely and adaptively increase their performance as the variety of samples available for learning becomes more.

4.4. System Architecture Description

This section provides detailed system architecture of Roboviz, overview of system modules, their structure and relationships are described in this section. User interfaces and related issues are also discussed.

4.4.1. Structure and Relationships

This section covers the overall technical description of Roboviz. It shows the working of application in perspective of different point-of-views and also shows relationship between different componets.

System Block Diagram

The diagram(s) show the higher level description of the application(s), generic working of the application(s) and interaction with the user.

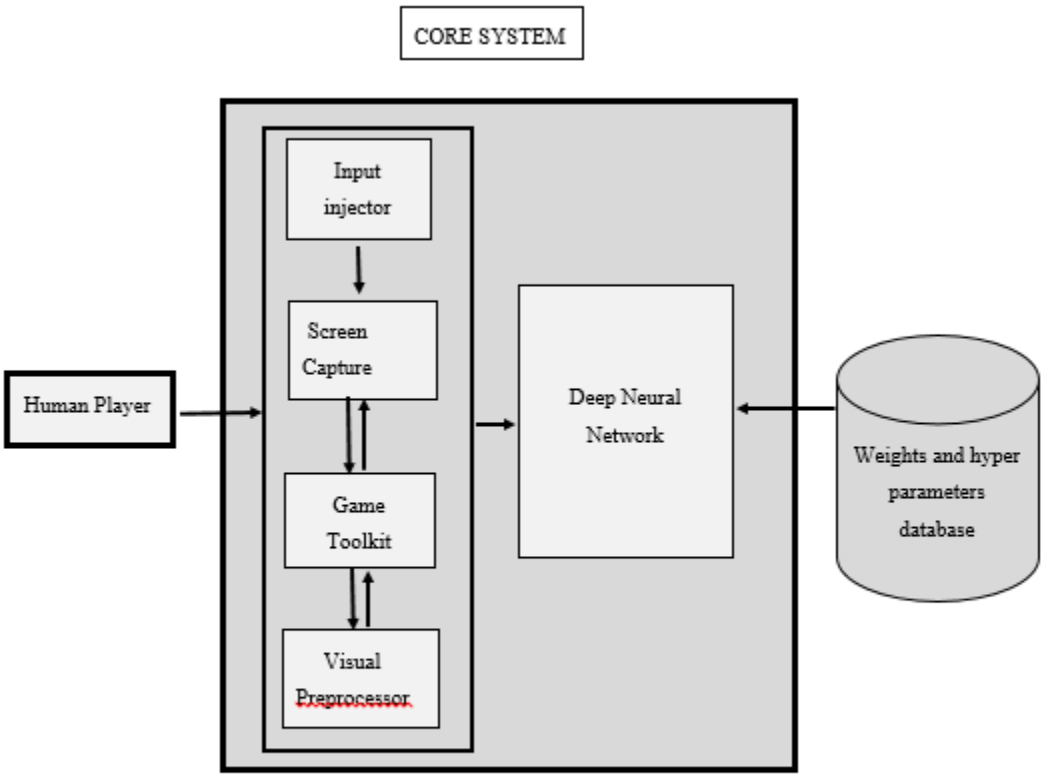

Figure 4-1 System Block Diagram

Abstract Diagram

The main components are

1. Game
2. Visual preprocessor
3. Screen Capture
4. Feature Analyzer
5. Input injector
6. Action evaluator

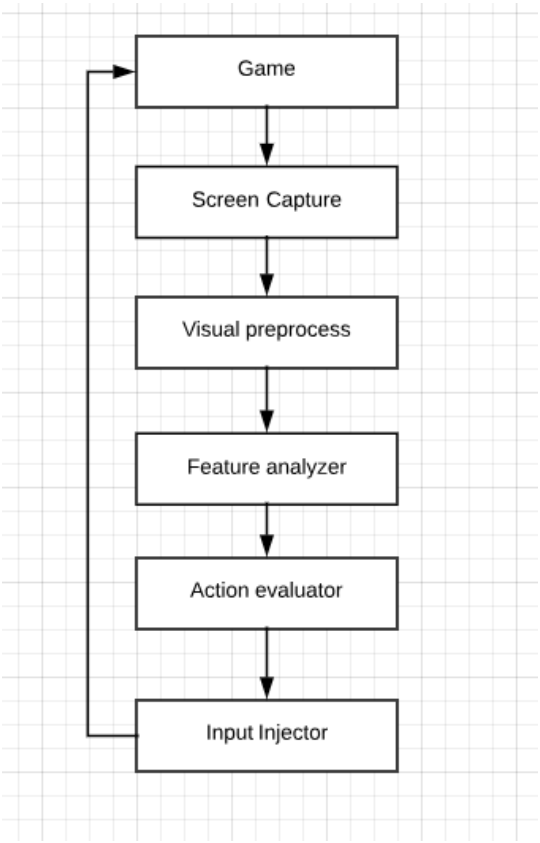

Figure 4-2 Component Diagram

User View (Use case diagram)

Figure 4-3 shows course of events that take place when an actor (user and other allowed

interactions) interact with the system.It shows the main functionaltiy of the application availaible for a normal user and how it interacts with those.

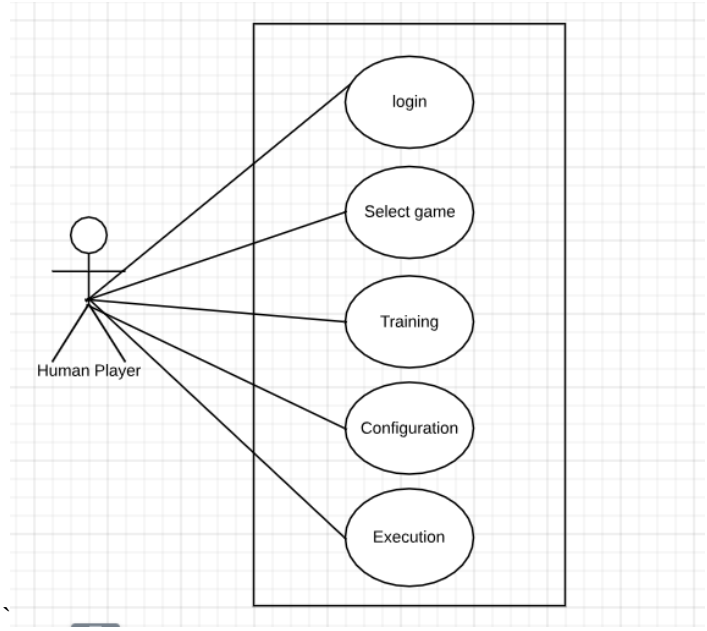

Figure 4-3 System Use Case Diagram

Actors: **Primary Actors**:

1. User(Player) **Secondary Actors:** None

Use Cases:

1. Login
2. Game selection
3. Training
4. Configuration
5. Execution

Use Case Description:

Use cases shown in the figure above are described below.

Use Case 1

Use Case ID:	1		
Use Case Name:	Login		
Actors:	Human player		
Created by:	Ahmad	Last Updated by:	Ahmad
Date Created:	07/01/2019	Date Last Updated:	07/01/2019
Description:	In this the human player will be able to log into the application with his/her username and password.		
Preconditions:	The player already has an account.		
Post conditions:	The player must be logged in		
Normal Flow (Primary Scenario):	1. Human will enter user name and password 2. The credentials will be verified from the database. 3.The access will be granted.		
Alternative Flows:	1. An error is encountered during login. 2. Proper functionality of the database will be checked.		

Table 4-1UseCase1

Use Case 2

Use Case ID:	2

Use Case Name:	Game selection		
Actors:	Human player		
Created by:	Ahmad	Last Updated by:	Ahmad
Date Created:	07/01/2019	Date Last Updated:	07/01/2019
Description:	A user will select the game to be played.		
Preconditions:	The user has been logged in.		
Post conditions:	The selected game would be opened to play		
Normal Flow (Primary Scenario):	1. The user will be logged in. 2. choice of games will be displayed 3.The game of choice will be selected		
Alternative Flows:	The error message will be generated.		

Table 4-2 UseCase2

Use Case 3

Use Case ID:	3		
Use Case Name:	Training		
Actors:	Human player		
Created by:	Ahmad	Last Updated by:	Ahmad
Date Created:	07/01/2019	Date Last Updated:	07/01/2019
Description:	This use case will help the agent to train itself iteratively by rewarding itself based on the goodness of the actions taken for a set of iterations.		
Preconditions:	Agent must be able to observe its environment. Proper action space should be provided on each state to the agent. Agent must be able to get reward based on the goodness of the actions taken.		
Post conditions:	Agent should be able to observe the environment after every action it takes. Agent should be able to calculate the estimated value (goodness) of its actions on each state based on its old experience. Agent should be able to use both exploration and exploitation (for re- estimation of values of its actions) techniques.		
Normal Flow (Primary Scenario):	The agent will born in an unknown environment and feed itself by choosing an action from the action space and then observing the changes that are made by its actions.		

Alternative Flows:	-

Table 4-3 UseCase 3

Use Case 4

Use Case ID:	4		
Use Case Name:	Configuration		
Actors:	Human user		
Created by:	Ahmad	Last Updated by:	Ahmad
Date Created:	07/01/2019	Date Last Updated:	07/01/2019
Description:	In this the neural network will be configured when the player plays against the bot.		
Preconditions:	User has to be logged in. Game has been selected..		
Post conditions:	Neural network will get configured		
Normal Flow (Primary Scenario):	1. bot learns from the player 2. network will be configured.		
Alternative Flows:	-		

Table 4-4 UseCase 4

Use Case 5

Use Case ID:	5

Use Case Name:	Execution		
Actors:	Human player		
Created by:	Ahmad	Last Updated by:	Ahmad
Date Created:	07/01/2019	Date Last Updated:	07/01/2019
Description:	The human player will be able to play against a trained bot		
Preconditions:	player has to log in.		
Post conditions:	-		
Normal Flow (Primary Scenario):	1. The bot will learn from the player 2. Bot will play against the human		
Alternative Flows:	-		

Table 4-5 UseCase 5

Sequence Diagrams

Following sequence diagrams show the sequence of activities performed in application.

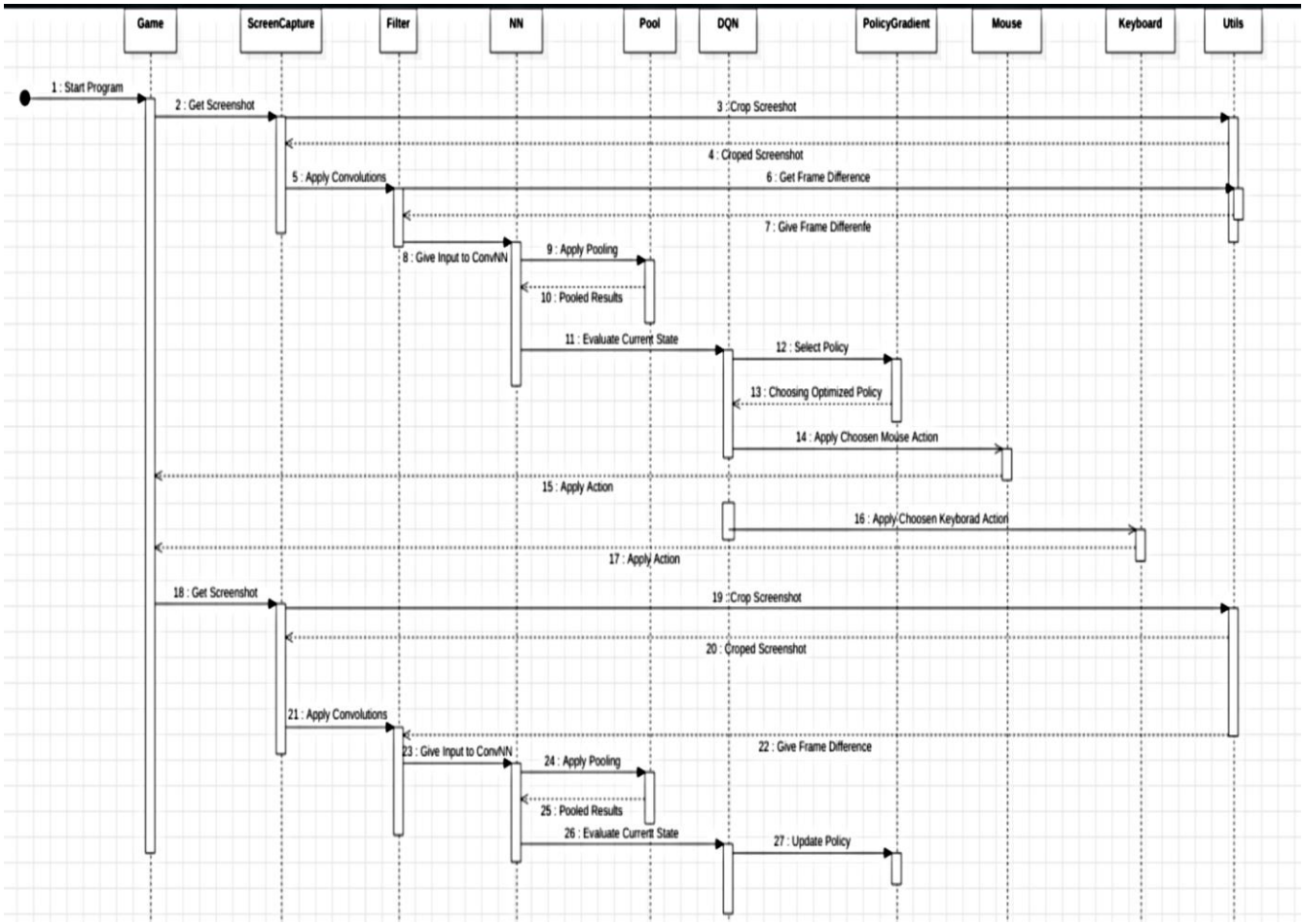

Figure 4-4 Sequence Diagram

Implementation View (Class Diagram)

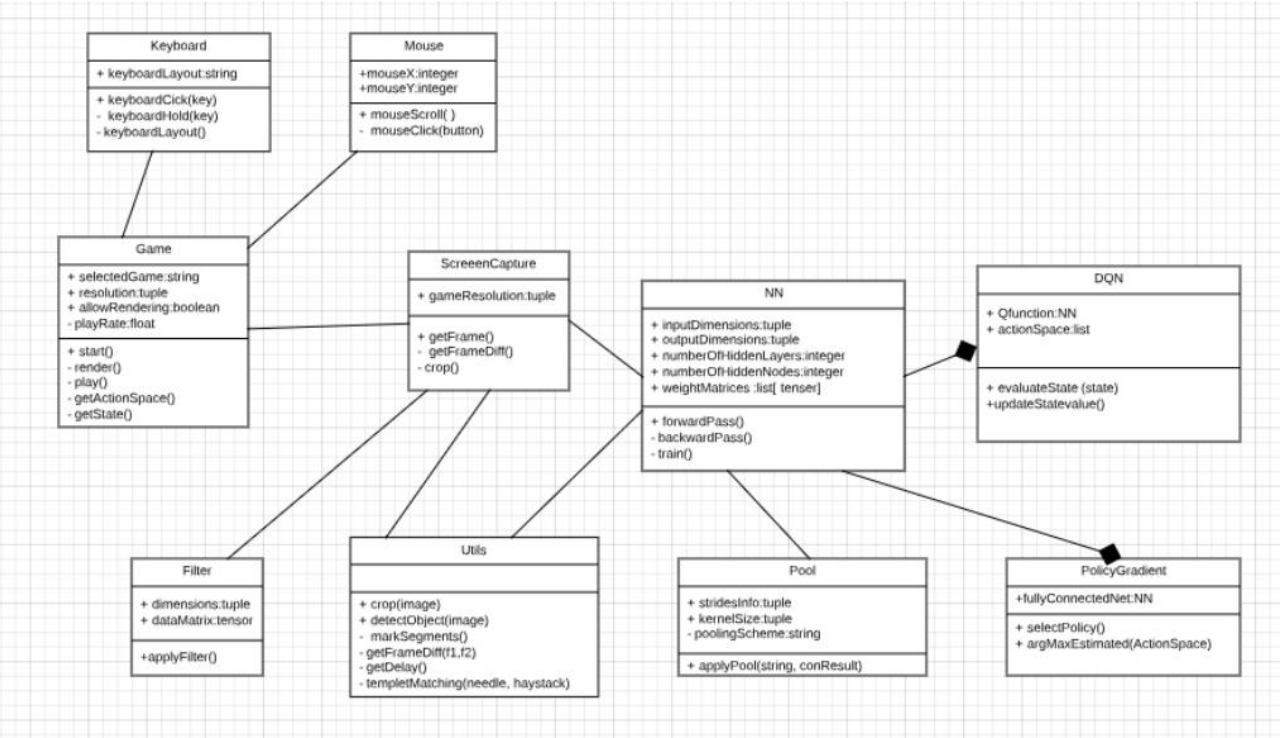

Figure 4-5 Class Diagram

Figure 4-12 Class Diagram

Class's description

Class Name	Description
Game	This class is responsible for the selection of the game. That is the selection of the learning environment. The other aim of this is rendering. Rendering or image synthesis is the automatic process of generating or creating a photorealistic or non-photorealistic image(environment) from a 2D or 3D model (or models in what altogether could be knows as a scene file) through computer programs. Also, the results of displaying or projecting such a model can be called a render. A scene file has objects in a specifically defined language or data structure; it would have geometry, viewpoint, texture, lighting, and shading information as a description

	of the virtual scene. The data present in the scene file is then sent to a rendering program to be processed and output to a digital image or raster graphics image file.
ScreenCapture	This class is responsible for all the functions related to frames per second of the game. They will include the functionalities like get Frame and get frame difference.
Visual PreProcessor	This class is responsible for the application of different filters on the images being rendered. This is used to extract features from imges. A convolution is an operation that converts a function into something else or something new. The purpose of convolutions is so that we can convert the original function into a form to extract more information. Convolutions have been adopted for ages in image processing to blur and sharpen images, and perform other operations, such as, enhance edges and emboss.
InputInjector	This class contains functions related to getting inputs from the human player through keyboard or mouse and translating them into meaningful actions in the simulated environment.
FeatureAnalyzer	This class contains the main neural network. The functions related to the learning of the bot will be present.

Table 4-6 Class's Description Dynamic View (Activity Diagram)

In activity diagram, the dynamic view of the system is shown. All the activities are shown.

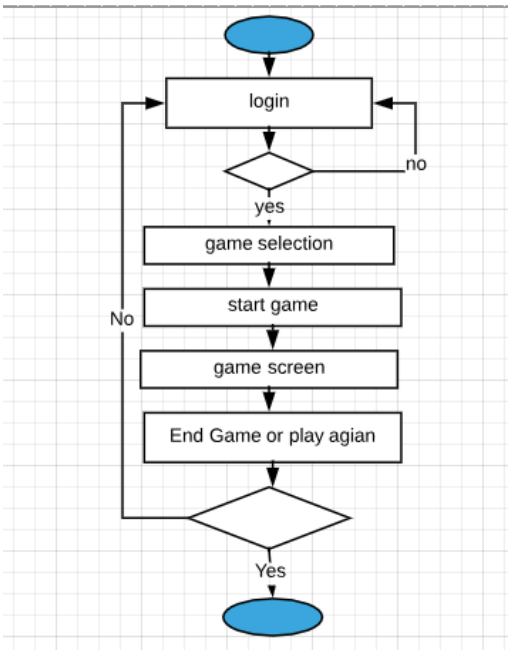

Figure 4-6 Activity Diagram

4.4.2 Detailed Description of Components

This section describes in detail all components of Roboviz application.

Game

Identification	**Game**
Type	Component
Purpose	This provides the simulation environment for the learning. Evaluate performance for each action in the game There can be multiple games.

Function	Start( ) Play() Render() getActionSpace() getState()
Subordinates	Graphic drivers, game engine
Dependencies	Rendering engine, physics engine
Interfaces	ApplyAction() GetResponse()
Resources	GPU, X64 compatible pc
Processing	Process physics Calculate positions Apply rigidbody dynamics Calculate rendering
Data	Textures , normal maps, characters models , bakes, animations.

Table 4-7 Game

Screen Capture

Identification	**Screen capture**
Type	Class
Purpose	Capture game screen every frame Crop to desired area.

Function	getScreenshot() Crop()
Subordinates	Game, screenshot utility, opencv context.
Dependencies	Opencv and operating system.
Interfaces	getCroppedResult() getRawScreenshot()
Resources	Rendered screen
Processing	gather raw screenshot data processed desired resolution output processed frame
Data	Raw screenshot , processed screenshot

Table 4-8 Screen Capture

Input Injector

Identification	**Input injector**
Type	Package
Purpose	To convert on-hot encoded vector that we got from action decider module into concrete user input with keyboard and mouse. Using user interface libraries such as pyautogui inject selected user input

Function	Convert (vector) Keyboard.click(button) Mouse.click(button) Keyboard.hold(button) Mouse.scroll(lines)
Subordinates	Keyboard and mouse drivers
Dependencies	Pyautogui
Interfaces	Gather one-hot encoded vector Inject user input
Resources	Python3 compatible platform
Processing	Convert one hot encoded vector to mouse and keyboard mappings Transfer keyboard and mouse action to drivers through graphical user input libraries such as pyautogui
Data	Data about System Administrator

Table 4-9 Input Injector

Action Decider

Identification	**Action decider**
Type	Algorithm
Purpose	Using the optimized deep learning techniques to learn the patterns between state space and action space. This will act as our policy considering as a classical reinforcement learning problem It will take convolution result as input and will output action probabilities

Function	Forward pass Backward pass Add hidden layer Set input output dimensions Loss function Optimizer
Subordinates	Linear algebra computational libraries bases on CUDA CuDNN
Dependencies	Keras pytorch
Interfaces	Gather result from convolution layers Output action probabilities
Resources	GPU , CUDA
Processing	In forward pass multiply weight matrices and input vectors apply activation function on the resulting vector this vector will now act as an input to the next hidden layer and this process will carry out till output layer. The output layer will transform the activations to a balanced probability distribution which will be further transformed into one-hot encoded vector
Data	Weight matrices , input vectors

Table 4-10 Action Decider

Feature Analyzer

Identification	**Feature analyzer**
Type	Utility
Purpose	To extract features from pre processed screenshots

	Apply multiple layers of convolutions augmented with activation and dropouts Use pooling to get the processed data into certain format Use flatten to convert 2D result into vectors
Function	ApplyConvolution() Activation() Dropout() Pool() Flatten()
Subordinates	Convolution utilities
Dependencies	Keras, pytorch
Interfaces	Send preprocessed image Get flattened vector result
Resources	GPU, Tensorflow or pytorch
Processing	Apply Convolution Apply activation function To make the processing more efficient and faster use GPU memory for storing matrix results Use dropouts to optimize
Data	Kernels, activation function parameters

Table 4-11 Feature Analyzer

Visual Preprocesor

Identification	**Visual preprocessor**

Roboviz

Type	Package
Purpose	To process game screens to desired format which is needed for feature analyzer in the next phase.
Function	ApplyFilter() convertFileFormat(destFormat)
Subordinates	ScreenCapture
Dependencies	Opencv
Interfaces	get processed result change desired format
Resources	Processed screenshot data, format rules , computer vision utilities
Processing	Apply filter to achieve desired image format Make the process efficient enough so that it can run every frame
Data	Screenshot

Table 4-12 Visual Preprocessor

4.5. Reuse and Relationships to other Products

Various applications exist that perform the functions of learning bots, but there will be very few applications that analyze the data and find meaningful reports out of it, that can not only tell us more about the data, but also shares the burden which was being performed manually.
The Atari games by google are provide us a very similar functionality .but they operate on very poor resolution. Our bot will operate in high resolution environment and will be multiplatform.

4.6. Design Decisions and Tradeoffs

The main tradeoff that exixts in this is that the computational power required for the processing is very high. The processing power required is very great and thus is computationally very expensive . unlike the existing Atari games our bot will work on high resolution.

MVC pattern will be used for the implementation of this application. General behavior of MVC is shown below.

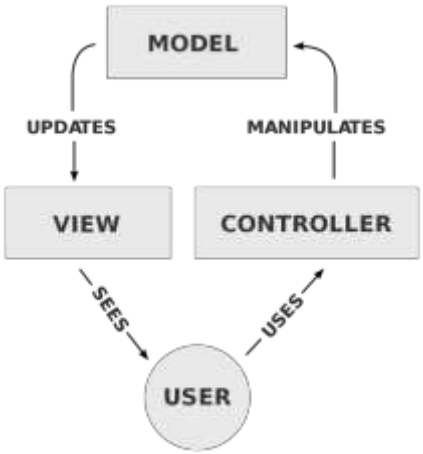

Figure 4-7 Architectural Diagram

Chapter 5 Project Test and Evaluation

5.1. Introduction

This test plan section describes the appropriate strategies, process and methodologies used to plan, execute and manage testing of the Roboviz. The test plan will ensure that Roboviz meets the customer requirements at an accredited level.

Manual Testing will be followed which includes testing a software manually, i.e., without using sthe software to identify any unexpected behavior or bug. Each Unit will be tested separately and then will be integrated with other units; therefore, Unit Testing and Integration testing will be followed. For each unit, Black box Testing is done and for combined units Acceptance Testing is done.

The test scope includes the Testing of all functional, application performance and use cases requirements listed in the *requirement document.*

Software testing, depending on the testing method employed, can be implemented at any time in the development process. However, most of the test effort occurs after the requirements have been defined and the coding process has been completed.

This document includes the plan, scope, approach and procedure of the testing of Roboviz. The pass/fail criteria of the test items are also defined. The document tracks the necessary information required to effectively define the approach to be used in the testing of the product.

5.2. Test Items

The test items selected for testing include the following

1. Account Management
2. General settings
3. Training

5.3. Features to Be Tested

The features of our game include the functionality mentioned in our design document. Following features are to be tested keeping in view the test items and system features aforementioned

1. Login.
2. Create account
3. Manage account
4. Remove users

5. Change passwords
6. Sound
7. Display
8. Resolution
9. Controls
10. Logout

5.4. Test Approach

Functional Testing will focus on each use case that is included in the version currently being worked on. Testing will mainly consist of execution of test cases written to address the gap identified. It will focus on inputs, outputs and system changes due to the actions. The testing strategy for Roboviz will be Alpha testing (Black box and White box techniques). Black Box testing technique will be used for testing functionality of each module.

5.5. Item Pass/Fail Criteria

Details of the test cases are specified in section Test Deliverables. Following the principles outlined below, a test item would be judged as pass or fail.

1. Preconditions are met
2. Inputs are carried out as specified
3. The result works as what specified in output => Pass
4. The system doesn't work or not the same as output specification => Fail.

5.6. Suspension Criteria and Resumption Requirements

Testing procedure will be suspended whenever a defect is found that restricts further testing. A corrective measure will be applied depending upon the criticality of the defect and testing will be resumed.
Efforts have been made to remove all and every chance of failure but there are certain unpredictable factors such as network issues, corrupt input data, or system failure that may lead to some issues. Error handling is applied more deeply to cover all these issues but unforeseen circumstances may happen.

5.7. Test Deliverables

Testing tasks

1. Develop Test Cases.
2. Execute tests based on the test cases developed.
3. Report defects during tests if any.
4. Manage the changes made after testing.

Test cases

Following are the Test Cases:

5.7.1. Application Startup

Test Case Name	Application Startup Testing
Test Case ID	1
Description	This feature sends the user to the login screen of the web application when he/she enters the website in the web browser.
Testing Technique Used	Black Box Testing

Preconditions	The computer is on and is connected to the internet, and a web browser is running.
Input Values	URL
Valid Inputs	The specified URL address
Steps	Enter URL in browser. Press Enter.
Expected Output	The user will be sent to the login screen of roboViz.
Actual Output	Successful opening of the application
Status	PASS

Table 5-1 TestCase 1

5.7.2 Login Feature Testing

Test Case Name	Login Feature Testing
Test Case ID	2
Description	This feature asks the user to enter his/her credentials for login. This test case is aimed to check that feature works according to user requirement.
Testing Technique Used	Black Box Testing
Preconditions	System is running and connected to database. User has opened the login webpage.
Input Values	Username Password
Valid Inputs	Valid and authorized username Valid and authorized password
Steps	Enter username. Enter password. Click “Login”.

Expected Output	The user credentials will be passed to the server for verification. The valid users will be directed to their dashboard after login.
Actual Output	Successful login. User is directed to the main application
Status	PASS

Table 5-2 TestCase 2

5.7.3. Register Account

Test Case Name	Register account
Test Case ID	3
Description	This test case checks the procedure of creation of user accounts.
Testing Technique Used	Black Box Testing
Preconditions	Application should be accessible to the user.
Input Values	Name, Email id, mobile number , password.
Valid Inputs	Alphanumeric values for the fields stated above
Steps	On login screen, click “Register”. Enter the data for the user. Click “Register”.
Expected Output	The new user is registered and added into the database and has access to the application.
Actual Output	The account is created in database. The user can access the application.
Status	PASS

Table 5-3 TestCase 3

5.7.4. Manage Account

Test Case Name	Manage Account
Test Case ID	4
Description	This feature allows the user to add or remove an account from the database.
Testing Technique Used	Black Box Testing
Preconditions	In case of removing an account the user must have an exsisting account.
Input Values	Account username
Valid Inputs	Account username
Steps	Click on the “manage accounts”.
Expected Output	The user will be able to add or delete account.
Actual Output	The user will be able to add or delete account
Status	PASS

Table 5-4 TestCase 4

5.7.5 Graphics configuration

Test Case Name	Graphics configuration
Test Case ID	5
Description	This test case checks if available settings related to graphics are displayed.
Testing Technique Used	Black Box Testing
Preconditions	User is logged in to the application. The game has been selected.
Input Values	Click on configuration.
Valid Inputs	Mouse Click
Steps	Click on “Configuration”
Expected Output	Settings options will be displayed.
Actual Output	Settings options will be displayed.
Status	PASS

Table 5-5 TestCase 5

5.7.6 Select Game

Test Case Name	Select game
Test Case ID	6
Description	This test case checks if the selected game runs properly.
Testing Technique Used	Black Box
Preconditions	The user is logged in the application. The game has been selected.
Input Values	Click on the game icon.
Valid Inputs	Mouse click on game icon.
Steps	Click on “game”.
Expected Output	The game session would initiate.
Actual Output	The game session would initiate.
Status	PASS

Table 5-6 TestCase 6

5.7.7. Game Controls

Test Case Name	Game Controls
Test Case ID	7
Description	This test case checks if the controls of the games can be selected according to the desire of the user.
Testing Technique Used	Black Box
Preconditions	The user is logged on to the application.

	The game has been selected.
Input Values	Mouse Click
Valid Inputs	Mouse Click
Steps	Click on “ Controls ”
Expected Output	The options to alter game controls will be displayed.
Actual Output	The options to alter game controls will be displayed
Status	PASS

Table 5-7 TestCase 7

5.7.8 Sound Check

Test Case Name	Sound check
Test Case ID	8
Description	This test case checks if the sound can be turned on and off or adjusted as desired.
Testing Technique Used	Black Box
Preconditions	The user is logged in the application. The user has selected the game.
Input Values	Mouse click
Valid Inputs	Mouse click
Steps	Login the application. Select game. Click on sound.
Expected Output	Sound will be adjusted, turned on and off as desired.
Actual Output	Sound will be adjusted, turned on and off as desired..

Status	PASS

Table 5-8 TestCase 8

5.7.9 Display

Test Case Name	Display
Test Case ID	9
Description	This test case checks if display configuration can be opened and adjusted properly.
Testing Technique Used	Black Box
Preconditions	The user is logged in. The game has been selected.
Input Values	Mouse click
Valid Inputs	Mouse click
Steps	Click on “display”.
Expected Output	The options for the adjustment and configuration for the display will be visible.
Actual Output	The options for the adjustment and configuration for the display will be visible.
Status	PASS

Table 5-9 TestCase 9

5.7.10. About

Test Case Name	About

Test Case ID	10
Description	This test case checks if the information about the application is displayed properly.
Testing Technique Used	Black Box
Preconditions	The user is logged on to the application.
Input Values	Mouse click
Valid Inputs	Mouse click
Steps	Click on “About” icon.
Expected Output	After the attacker clicks on about icon the information about the application is displayed.
Actual Output	After the attacker clicks on about icon the information about the application is displayed.
Status	PASS

Table 5-10 TestCase 10

5.7.11. Log out

Test Case Name	Logout
Test Case ID	12
Description	This test case checks if the user is able to logout properly.
Testing Technique Used	Black Box
Preconditions	The user must be logged in the application.
Input Values	Mouse click
Valid Inputs	Mouse click

Steps	Click on logout icon
Expected Output	The user will be logged out of the application and directed to the login page.
Actual Output	The user will be logged out of the application and directed to the login page
Status	PASS

Table 5-11 TestCase 11

5.8. Responsibilities, Staffing and Training Needs

5.8.1. Responsibilities:

All developers of the project are responsible for the completion of all components testing and integration testing tasks.

5.8.2. Staffing and Training Needs:

Basics knowledge of testing strategies and techniques is needed for the testing of the project. Techniques such as Black Box testing, integration testing should be known to developers.

All the developers will be testing each other's work and will be actively participating in the development and testing of the project simultaneously.

5.9. Risk and Contingencies

Efforts have been made to remove all and every chance of failure but there are certain unpredictable factors such as network issues, corrupt input data, or system failure that may lead to some issues. Error handling will be applied more deeply to cover all these issues but unforeseen circumstances may happen.

5.9.1. Schedule Risk:

The project might get behind schedule so in order to complete the project in time we will be needing to increase the haours/day that the project is being worked on.

5.9.2. Operational Risks:

Operational risks will be eliminated by Scheduling daily meetings and regular deadlines to meet the goals of the project as well as provide proper communication within the group.

5.9.3. Technical risks:

Technical risks will be eliminated by keeping the once defined requirements constant.

5.9.4 Programmatic Risks:

In case of a programmatic risk the scope of the project will be limited in order to stay inside the constraints of the project.

Chapter 6 Future Work

This project can be extended further for future support of dyslexia patients

1. Learning from 3D models.
2. Multi platform bot.
3. Games with high resolution.
4. Analytics and report generation on gathered data.
5. Reinforcement learning in multiple gaming environments..

Chapter 7 Conclusion

Many of the practical things around us need persistent practice before it can be perfected.
With practice comes perfection, but each trial and error has its own cost and takes time. Sometimes that cost is too much and cannot be paid or is not feasible to be paid. There are cases where something needs to be learned quickly but cannot be done without all the practice. All can be solved using a simulated environment.

This project is intended to provide a deep neural network model using a variety of machine learning techniques, including computer vision, reinforcement tree learning, and case-based reasoning. We are implementing these techniques to form a novel approach for training AI-bot on its gaming environment. Bot will use computational methods to "learn" information directly from data without relying on a predetermined equation as a model. The algorithms adaptively improve their performance as the number of samples available for learning increases.
The product is a project that will use "Keras library" of neural network model to train model using different reinforcement techniques. Objects will be recognized through computer vision. Game will be developed using gaming development toolkit. Training API's will train the model by using GPU computational power. Bot will be added to the game..

Appendices

Appendix A: Glossary

1. **APP:** Application
2. **GUI:** Graphical User Interface
3. **GPU:** Graphic processing unit.
4. **SDS**: Software Design Specification
5. **UML**: The Unified Modeling Language

Bibliography

Bibliography

https://www.smithsonianmag.com/innovation/google-ai-deepminds-alphazero-games-chess-and-go-180970981/

https://www.atari.com/about-us/

https://arxiv.org/abs/1312.5602

https://www.sciencedirect.com/topics/neuroscience/reinforcement-learning

Reinforcement Learning: A Tutorial Survey and Recent Advances Abhijit Gosavi Department of Engineering Management and Systems Engineering 219 Engineering Management Missouri University of Science and Technology Rolla, MO 65409

http://www.academia.edu/Documents/in/Reinforcement_Learning

A Concise Introduction to Reinforcement Learning Ahmad Hammoudeh RLR1, Data Science programme2 AIC1, Princess Suamaya University for Technology2 Amman, Jordan

Deep Reinforcement Learning: An OverviewSeyed Sajad Mousavia, Michael Schukata, Enda HowleyaaThe College of Engineering and InformaticsNational University of Ireland, Galway, Republic of Ireland

IEEE SIGNAL PROCESSING MAGAZINE, SPECIAL ISSUE ON DEEP LEARNING FOR IMAGE UNDERSTANDING (ARXIV EXTENDED VERSION) 1
A Brief Survey of Deep Reinforcement Learning
Kai Arulkumaran, Marc Peter Deisenroth, Miles Brundage, Anil Anthony Bharath

Reinforcement Learning: A Survey Leslie Pack Kaelbling lpk@cs.brown.edu Michael L. Littman mlittman@cs.brown.edu Computer Science Department, Box 1910, Brown University Providence, RI 02912-1910 USA Andrew W. Moore awm@cs.cmu.edu Smith Hal l 221, Carnegie Mel lon University, 5000 Forbes Avenue Pittsburgh, PA 15213 USA

Roboviz

by Ibrahim Khan

Submission dat e: 17- May- 2019 05:26AM (UT C+0500)
Submission ID: 1131691962
File name: 84712_Ibrahim__Khan_Roboviz_510008_738724983.docx (1.35M)
Word count : 7346
Charact er count : 39207

Roboviz

ORIGINALITY REPORT

14%	7%	2%	13%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to University of Melbourne Student Paper	2%
2	Submitted to Misr International University Student Paper	1%
3	www.saama.com Internet Source	1%
4	Submitted to University of Maryland, University College Student Paper	1%
5	Robin Baumgarten. "Combining AI Methods for Learning Bots in a Real-Time Strategy Game", International Journal of Computer Games Technology, 2009 Publication	1%
6	Submitted to Higher Education Commission Pakistan Student Paper	1%
7	Submitted to University of Malaya Student Paper	1%

8	www.slideshare.net Int ernet Source	1%
9	Submitted to Universiti Tunku Abdul Rahman St udent Paper	1%
10	ibimapublishing.com Int ernet Source	1%
11	www.mathworks.com Int ernet Source	1%
12	Submitted to Majan College St udent Paper	<1%
13	Submitted to GLA University St udent Paper	<1%
14	Submitted to University of Birmingham St udent Paper	<1%
15	www.scribd.com Int ernet Source	<1%
16	Submitted to Columbia College of Missouri St udent Paper	<1%
17	於2012-06-13提交至Higher Education Commission Pakistan St udent Paper	<1%
18	hep-proj-grid-fabric.web.cern.ch Int ernet Source	<1%

19	Submitted to Liverpool Media Academy St udent Paper	<1%
20	Submitted to Pathfinder Enterprises St udent Paper	<1%
21	Submitted to University of Greenwich St udent Paper	<1%
22	dacg.in Int ernet Source	<1%
23	www.studymode.com Int ernet Source	<1%
24	Submitted to London School of Commerce St udent Paper	<1%
25	Submitted to University of Auckland St udent Paper	<1%
26	Submitted to Colorado Technical University Online St udent Paper	<1%
27	Submitted to University of Bradford St udent Paper	
28	Submitted to Monmouth University St udent Paper	<1%
29	Submitted to University of Central Lancashire St udent Paper	<1%
		<1%

30	Roboviz Submitted to CSU, San Diego State University Student Paper	<1%
31	web3.cc.utexas.edu Internet Source	<1%
32	Submitted to University of Westminster Student Paper	<1%
33	Submitted to University of London External System Student Paper	<1%

Exclude quotes On Exclude bibliography On

Exclude matches	Off