

Mini Project report on

ANTISTRESS GAME

By

DIVYA SUNDARAM [2020510062]

RASHIKA SUVARNA [2020510063]

Under the guidance of

Internal Supervisor

[Prof. Sakina Salmani]



Department of Master of Computer Applications

Sardar Patel Institute of Technology

Autonomous Institute Affiliated to Mumbai University

2020-21

CERTIFICATE OF APPROVAL

This is to certify that the following student

DIVYA SUNDARAM [2020510062]

RASHIKA SUVARNA [2020510063]

Have satisfactorily carried out work on the project entitled

“ANTISTRESS GAME”

Towards the fulfilment of mini project during the year 2020-21

Project Guide

(Prof. Sakina Salmani)

PROJECT APPROVAL CERTIFICATE

This is to certify that the following students

DIVYA SUNDARAM [2020510062]

RASHIKA SUVARNA [2020510063]

Have successfully completed the Project report on “ANTISTRESS GAME”, which is
found to
be satisfactory and is approved

At

SARDAR PATEL INSTITUTE OF TECHNOLOGY,
ANDHERI (W), MUMBAI.

INTERNAL EXAMINER

EXTERNAL EXAMINER

Head of Department

(Dr. Pooja Raundale)

Principal

(Dr. B. N. Chaudhari)

INDEX

Contents

Abstract.....	6
Objectives.....	7
List of Figures	8
1. Introduction	9
1.1 Problem Definition.....	9
1.2 Objectives and Scope.....	9
1.3 Existing System	9
1.4 Proposed System.....	10
1.5 System Requirements	10
2. Literature Survey	11
3. SRS and Design.....	12
3.1 Introduction.....	12
3.1.1 Purpose.....	12
3.1.2 System Overview	12
3.2 Overall Design.....	12
3.2.1 Product Functions	12
4. Project Analysis and Design.....	13
4.1 Methodologies Accepted.....	13
4.1.1 Detailed life cycle of Project	13
4.2 UML Diagrams	15
4.2.1 Use case Diagram	15
4.2.2 Activity Diagram	16
5. Project Implementation and Testing.....	18
5.1 Gantt Chart.....	18
5.2 Pert Chart	19
5.3 Timeline chart.....	20
5.4 Code with reference to design.....	21

5.5 Snapshot of UI.....	23
6. Future Enhancements	29
7. Limitations	30
9. Conclusion.....	31
10. Bibliography	32

Abstract

Stress is the body's reaction to pressure from a certain situation or event. It can be a physical, mental, or emotional reaction.

Usually, people scroll through social media or play games in short breaks. But losing a game adds to more stress. In order to avoid this, we wanted to make a game to increase the positivity in a person.

According to research, games like bubble wrap, piano and block puzzles have been proven to be the most effective ways of helping to reduce stress levels.

There is one such game called Antistress in the Playstore.

Objectives

- · The app can be run on any basic android application.
- · Stress reliever games reduce depression.
- · To give a calmer state of mind.
- · We might not have access to physical antistress toys all the time, but have mobile phones during commuting etc.

List of Figures

Figure 4.1: Spiral Model	14
Figure 4.2: Use case diagram	16
Figure 4.3: Activity Diagram	17
Figure 5.1: Gantt Chart	18
Figure 5.2: Gantt Chart table	19
Figure 5.3: Pert Chart	20
Figure 5.4: Timeline chart.....	21
Figure 5.5: Splash Screen1.....	23
Figure 5.6: Splash Screen 2.....	23
Figure 5.7: Splash Screen 3	24
Figure 5.8: Menu page.....	24
Figure 5.9: Block puzzle.....	25
Figure 5.10: Block puzzle game.....	25
Figure 5.11: Bubble game.....	27
Figure 5.12: Piano Game.....	28

1. Introduction

1.1 Problem Definition

Work life gives a lot of stress and anxiety. People usually spend a lot of time commuting to work and get a small break in between. And they usually choose to play games, but many times, losing a game or the fear of losing a game or the wish to win a game gives anxiety and we get addicted to it and lose track of our time.

1.2 Objectives and Scope

Our aim was to make a game which will reduce the stress level.

The app has three antistress games in total.

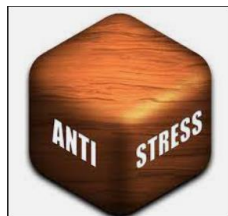
1. **Block Puzzle game:** Drag and drop blocks of different shapes to the grid. When a line in the grid is filled with the blocks, then it is removed accordingly.
2. **Bubble pop:** The screen is populated by bubbles. On touching a bubble, it popped. When all the bubbles in the screen are popped, a positive affirmation is shown on the screen to improve the mood of the user.
3. **Piano:** A virtual piano is made.

1.3 Existing System

1. Bubble wrap:

- Visual and sound effects on popping a bubble.
- The bubbles reset when all the bubbles are popped.
- A congratulatory voice can be heard, once all the bubbles are popped.

2. Blocks:



- We can just replace the blocks of different shapes and sound can be heard accordingly.

1.4 Proposed System

1. Adding quotes in bubble wrap game.
2. Converting the block game into a block puzzle game.
3. Building a virtual piano. The different sound effects can be generated.

1.5 System Requirements

Smartphone with minimum Android version 4.1.

2. Literature Survey

Stress is a feeling of emotional or physical tension. It can come from any event or thought that makes you feel frustrated, angry, or nervous.

Stress is your body's reaction to a challenge or demand. In short bursts, stress can be positive, such as when it helps you avoid danger or meet a deadline. But when stress lasts for a long time, it may harm your health.

Many other studies have shown that engaging in activities helps reduce stress levels and improve mood, while sedentary behavior may lead to increased stress, poor mood, and sleep disturbances.

3. SRS and Design

3.1 Introduction

3.1.1 Purpose

The purpose of this application is to provide a collection of games in one place, which helps the user to pass the time, have fun and in turn help in reducing the stress of the user.

3.1.2 System Overview

The application has a menu page with an antistress music associated with it. The page has three buttons each leading to another game, the user can choose the game which he/she wishes to play.

3.2 Overall Design

3.2.1 Product Functions

Stress is a common issue found in today's generation. The app aims to reduce stress in the form of a game, which can be played to break the monotonous routine. The application has been built in Unity with C#.

4. Project Analysis and Design

4.1 Methodologies Accepted

4.1.1 Detailed life cycle of Project

Project Management is the process of scoping, planning, organizing and controlling the development of an acceptable system at a minimum cost and within a specified time frame.

Choosing a model: Among the various popular models for developing a model, we have chosen Spiral model.

The *spiral model* combines the idea of iterative development with the systematic, controlled aspects of the waterfall model. Spiral model is a combination of iterative development process model and sequential linear development model i.e. waterfall model with very high emphasis on risk analysis. It allows for incremental releases of the product, or incremental refinement through each iteration around the spiral.

The spiral model has four phases. A software project repeatedly passes through these phases in iterations called Spirals.

-Identification: This phase starts with gathering the business requirements in the baseline spiral. In the subsequent spirals as the product matures, identification of system requirements, subsystem requirements and unit requirements are all done in this phase. This also includes understanding the system requirements by continuous communication between the customer and the system analyst. At the end of the spiral the product is deployed in the identified market.

-Design: Design phase starts with the conceptual design in the baseline spiral and involves architectural design, logical design of modules, physical product design and final design in the subsequent spirals.

-Construct or Build: Construct phase refers to production of the actual software product at every spiral. In the baseline spiral when the product is just thought of and the design is being developed a POC (Proof of Concept) is developed in this phase to get customer feedback. Then in the subsequent spirals with higher clarity on requirements and design details a working model of the

software called build is produced with a version number. These builds are sent to customer for feedback.

-Evaluation and Risk Analysis: Risk Analysis includes identifying, estimating, and monitoring technical feasibility and management risks, such as schedule slippage and cost overrun. After testing the build, at the end of first iteration, the customer evaluates the software and provides feedback.

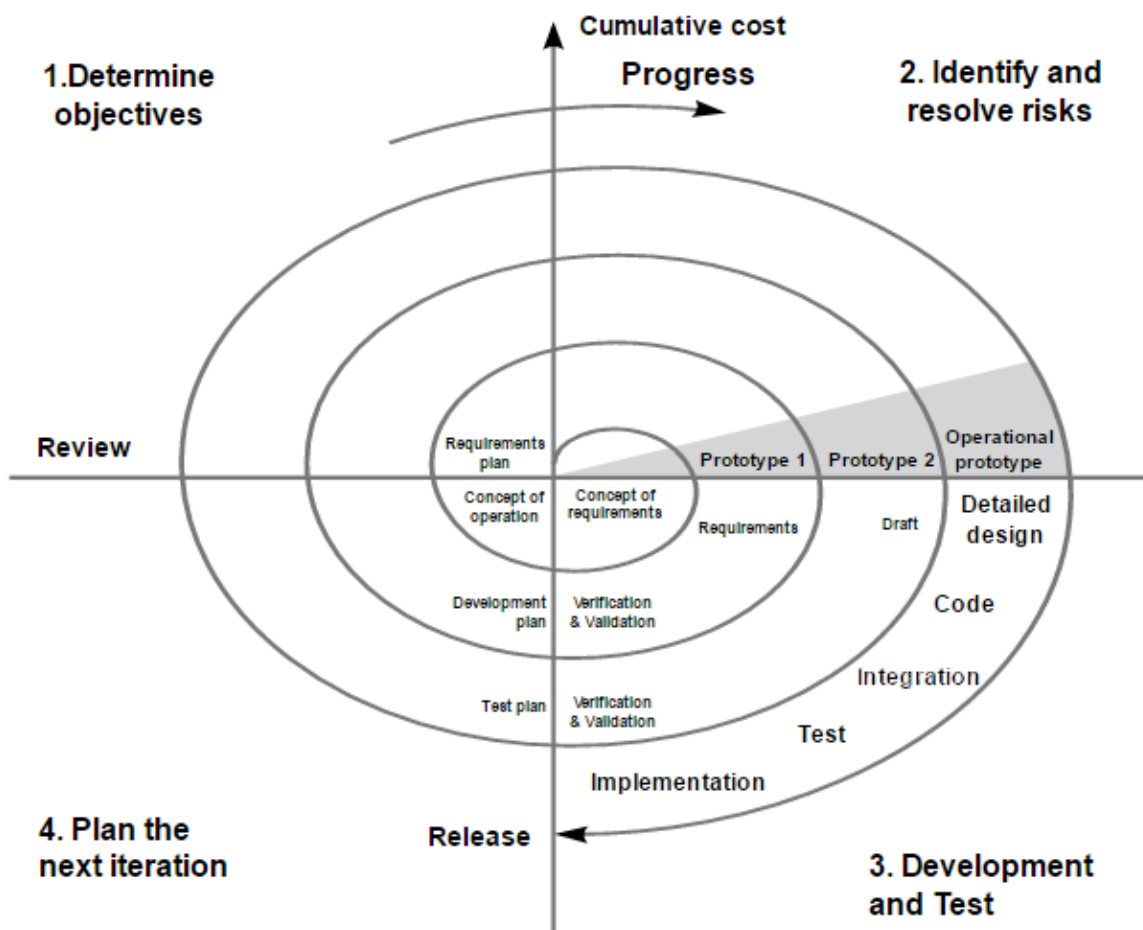


Figure 4.1 : Spiral Model

Advantages of spiral model:

- Changing requirements can be accommodated.
- Allows for extensive use of prototypes.
- Requirements can be captured more accurately.
- Users see the system early.
- Development can be divided into smaller parts and more risky parts can be developed earlier which helps better risk management.

When to use the spiral model:

- When costs there is a budget constraint and risk evaluation is important.
- For medium to high-risk projects.
- Customer is not sure of their requirements which is usually the case.
- Requirements are complex and need evaluation to get clarity.
- New product line which may be released in phases to get enough customer feedback.
- Significant changes are expected in the product during the development cycle.

4.2 UML Diagrams**4.2.1 Use case Diagram**

Use case diagrams are a set of use cases, actors and their relationships. They represent the use case view of a system. A use case represents a particular functionality of a system. So a use case diagram is used to describe the relationships among the functionalities and their internal/external controllers. These controllers are known as actors.

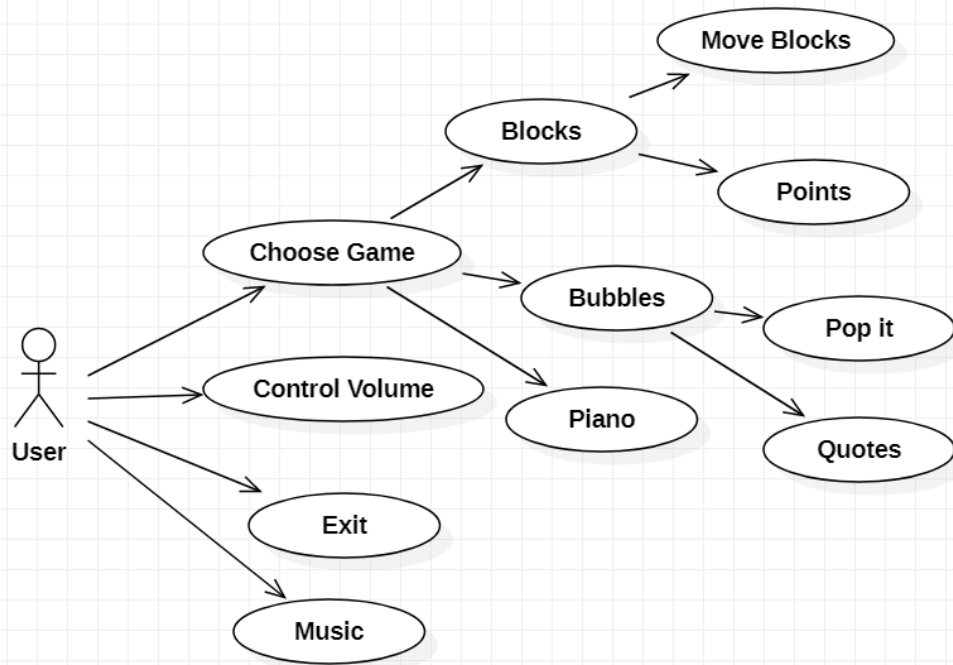


Figure 4.2: Use case diagram

4.2.2 Activity Diagram

Activity diagram is another important diagram in UML to describe dynamic aspects of the system. Activity is a particular operation of the system. Activity diagrams are not only used for visualizing the dynamic nature of a system but they are also used to construct the executable system by using forward and reverse engineering techniques. The only missing thing in the activity diagram is the message part. It does not show any message flow from one activity to another. Activity diagram is sometimes considered as the flow chart. Although the diagram looks like a flow chart, it is not. It shows different flows like parallel, branched, concurrent and single.

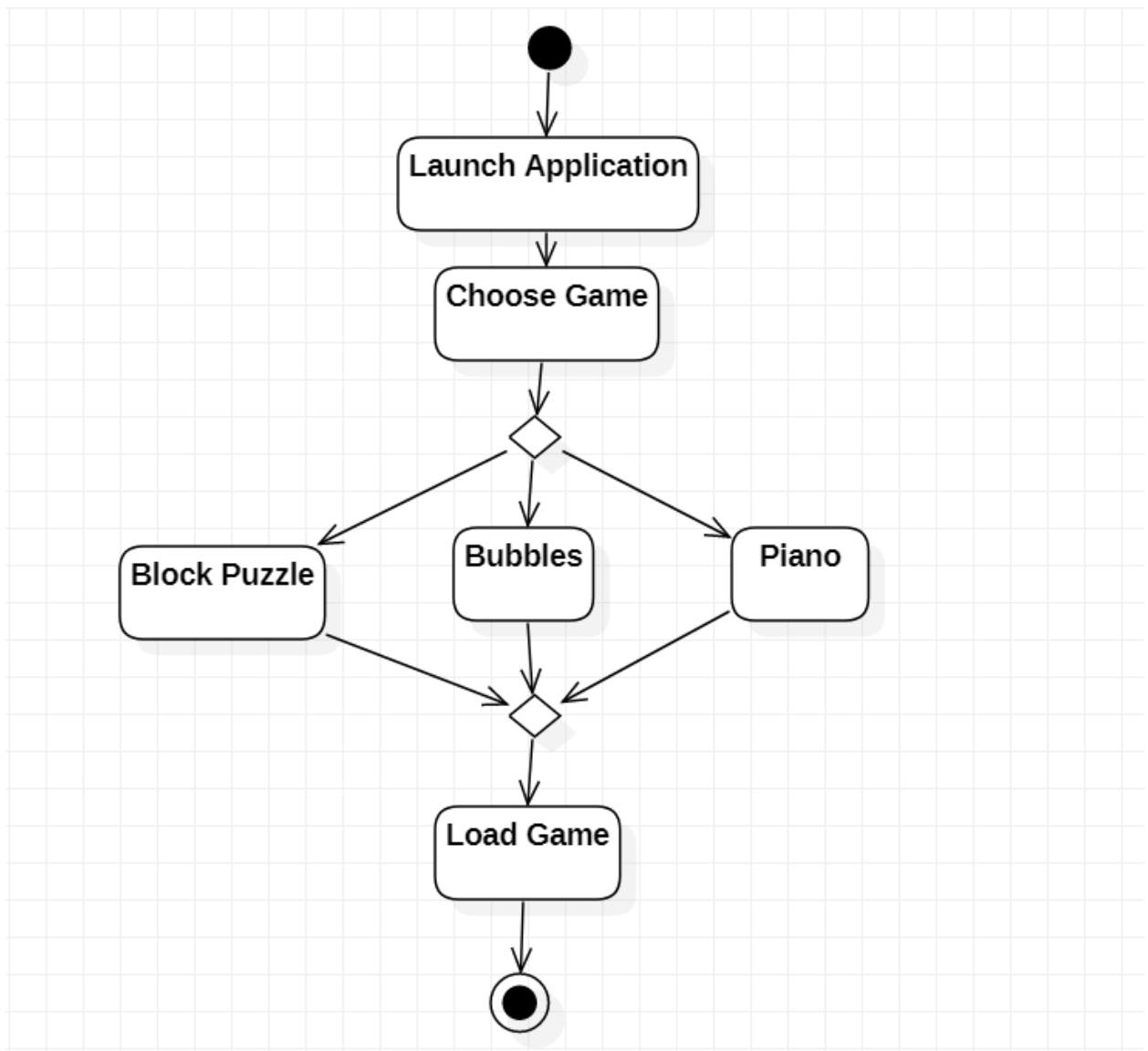


Figure 4.3: Activity Diagram

5. Project Implementation and Testing

5.1 Gantt Chart

Gantt chart is a type of a bar chart that is used for illustrating project schedules. Gantt charts can be used in any projects that involve effort, resources, milestones and deliveries. Gantt charts allow project managers to track the progress of the entire project. Through Gantt charts, the project manager can keep a track of the individual tasks as well as of the overall project progression. Gantt charts can be successfully used in projects of any scale. When using Gantt charts for large projects, there can be an increased complexity when tracking the tasks.

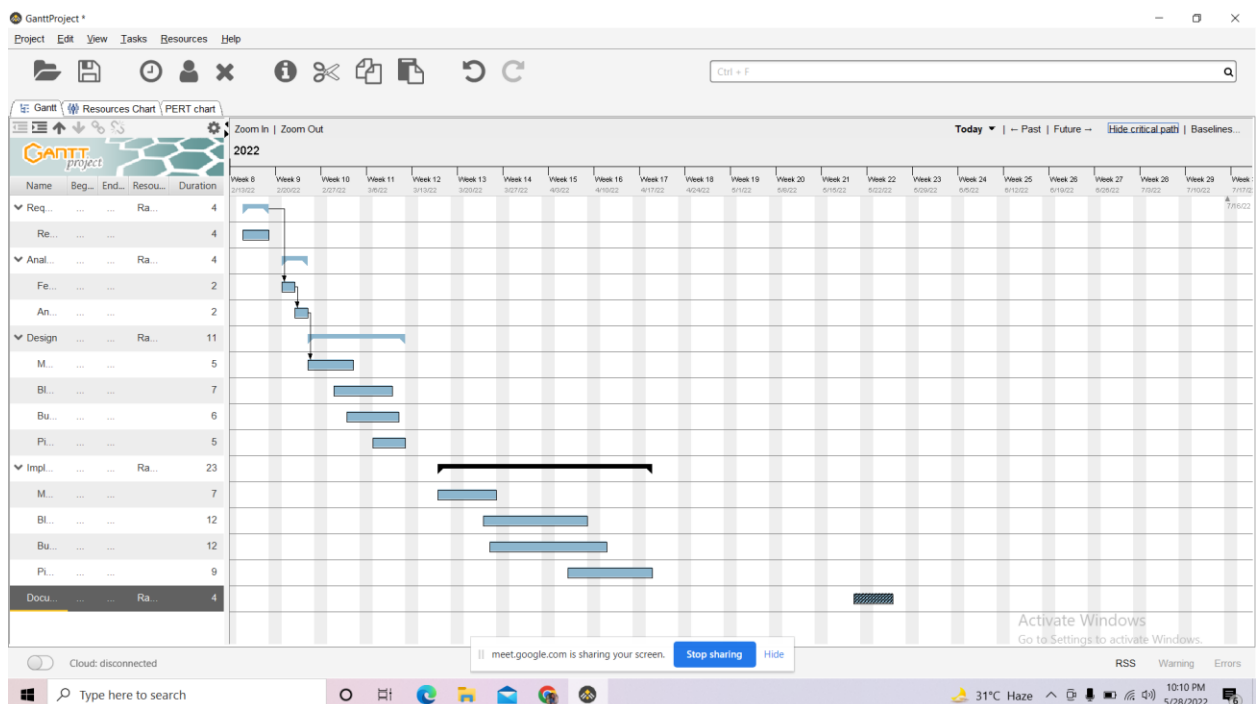


Figure 5.1: Gantt Chart

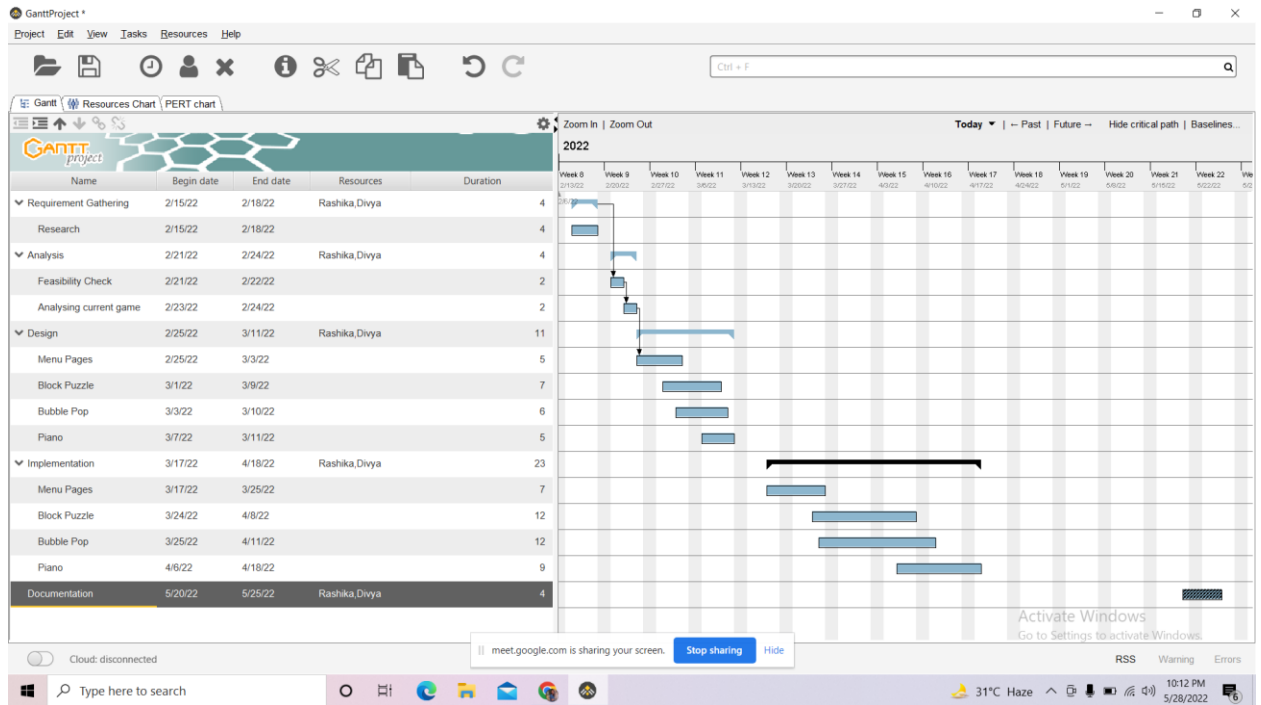


Figure 5.2: Gantt Chart – table

5.2 Pert Chart

A PERT chart presents a graphic illustration of a project as a network diagram consisting of numbered nodes (either circles or rectangles) representing events, or milestones in the project linked by labelled vectors (directional lines) representing tasks in the project. The direction of the arrows on the lines indicates the sequence of tasks. The PERT chart is sometimes preferred over the Gantt chart, another popular project management charting method, because it clearly illustrates task dependencies. On the other hand, the PERT chart can be much more difficult to interpret, especially on complex projects.

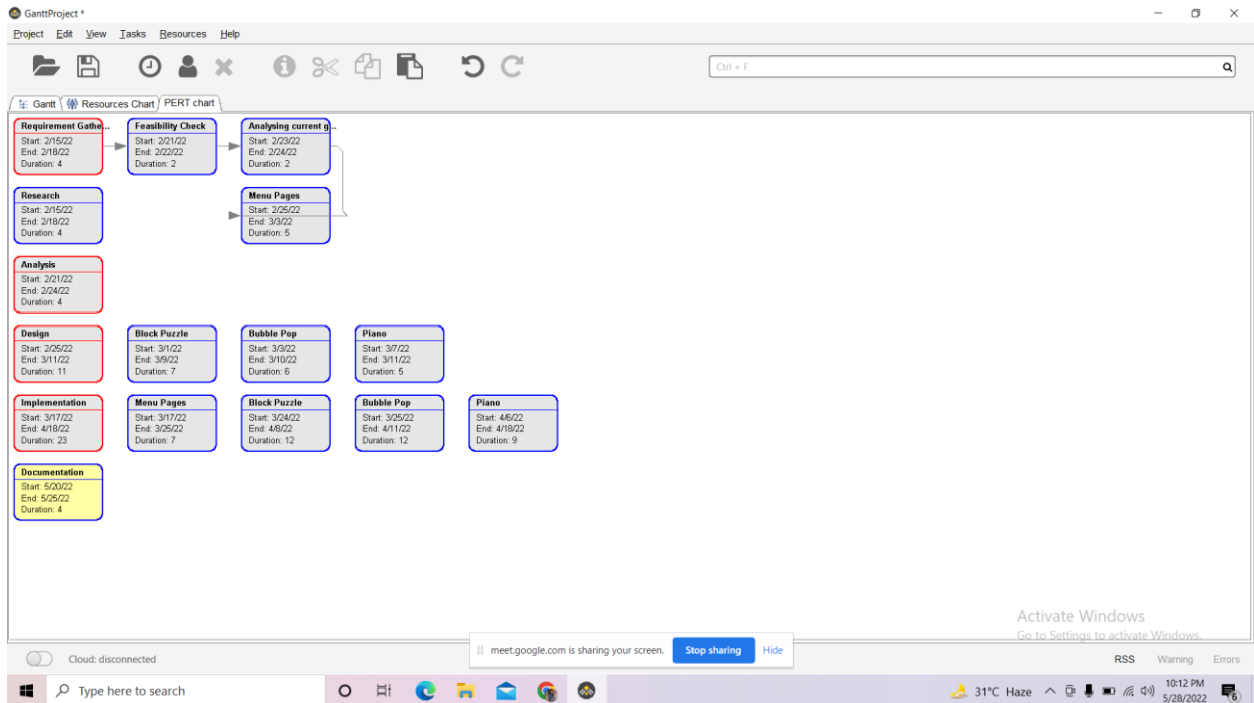


Figure 5.3: Pert Chart

5.3 Timeline chart

A timeline is a way of displaying a list of events in chronological order, sometimes described as a project artifact. It is typically a graphic design showing a long bar labelled with dates alongside itself and usually events labelled on points where they would have happened. Timelines can use any time scale, depending on the subject and data. Most timelines use a linear scale, where a unit of distance is equal to a set amount of time. This time scale is dependent on the events in the timeline. There are many methods of visualizations for timelines. Historically, timelines were static images, and generally drawn or printed on paper. Timelines relied heavily on graphic design, and the ability of the artist to visualize the data.

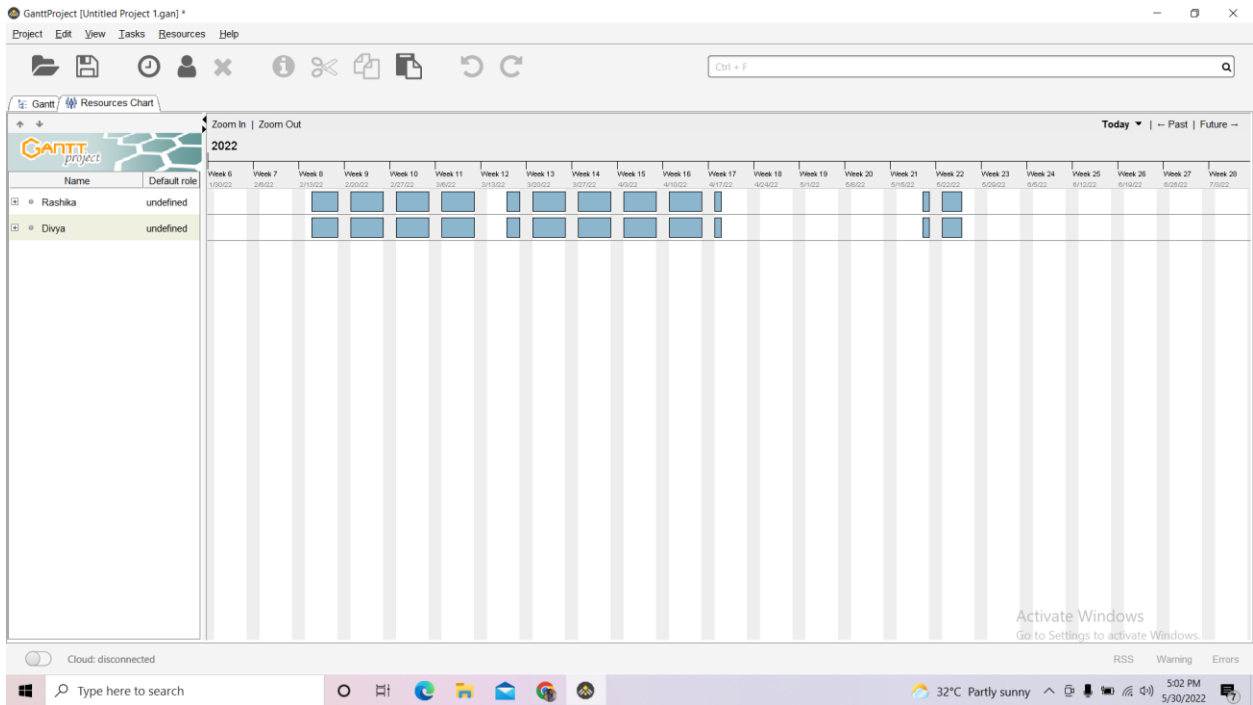


Figure 5.4: Timeline chart

5.4 Code with reference to design

SceneM.cs

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.SceneManagement;
using EasyUI.Dialogs;
using UnityEngine.UI;
using System.Linq;

public class SceneM : MonoBehaviour
{
    private static bool m_Initialized = false;
    void Start()
    {
        if (!m_Initialized)
        {
            m_Initialized = true;
            // First Dialog -----
            DialogUI.Instance
                .SetMessage ( "Hey, Welcome! Stressed enough?? Don't worry, We have got you covered:)" )
        }
    }
}
```

```

// .SetButtonColor ( DialogButtonColor.Blue )
.SetButtonText("Next")
.OnClose ( () => Debug.Log ( "Closed 1" ) )
.Show ( );

// Second Dialog -----
DialogUI.Instance
.SetMessage ( "When you need relaxation, diversion or just a moment of distraction enjoy this
collection of games!" )
// .SetButtonColor ( DialogButtonColor.Magenta )
.SetButtonText ( "Next" )
.OnClose ( () => Debug.Log ( "Closed 2" ) )
.Show ( );

// Third Dialog -----
DialogUI.Instance
.SetMessage ( "Let's get Started!, Hope you burst your stress..ENJOY!!!" )
.SetFadeInDuration ( 1f )
// .SetButtonColor ( DialogButtonColor.Red )
.SetButtonText("Start Game!")
.OnClose ( () => Debug.Log ( "Closed 3" ) )
.Show ( );
}
}

public void OpenBubble() {
    SceneManager.LoadScene(SceneManager.GetActiveScene().buildIndex +1 );
}
public void OpenPiano() {
    SceneManager.LoadScene(SceneManager.GetActiveScene().buildIndex +2 );
}
public void OpenBlock() {
    SceneManager.LoadScene("MainMenu");
}
}

```

5.5 Snapshot of UI

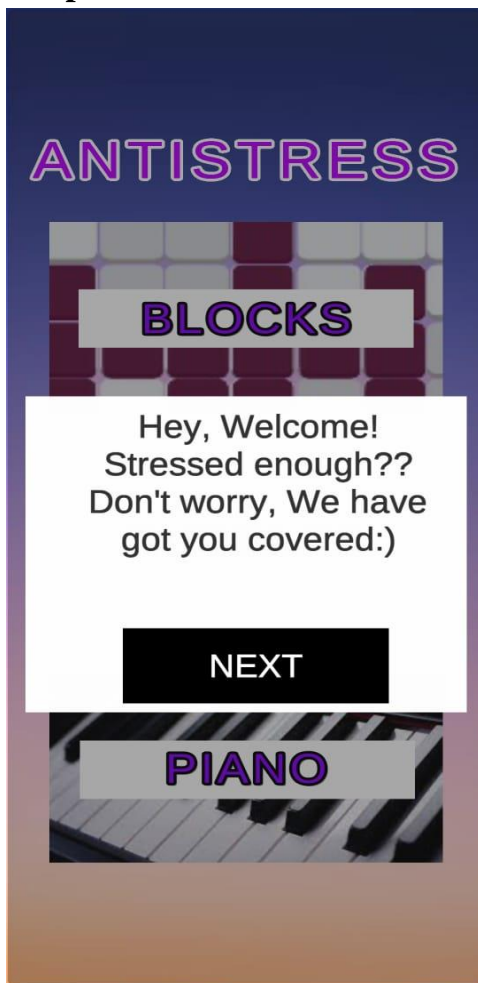


Figure 5.5: Splash Screen1

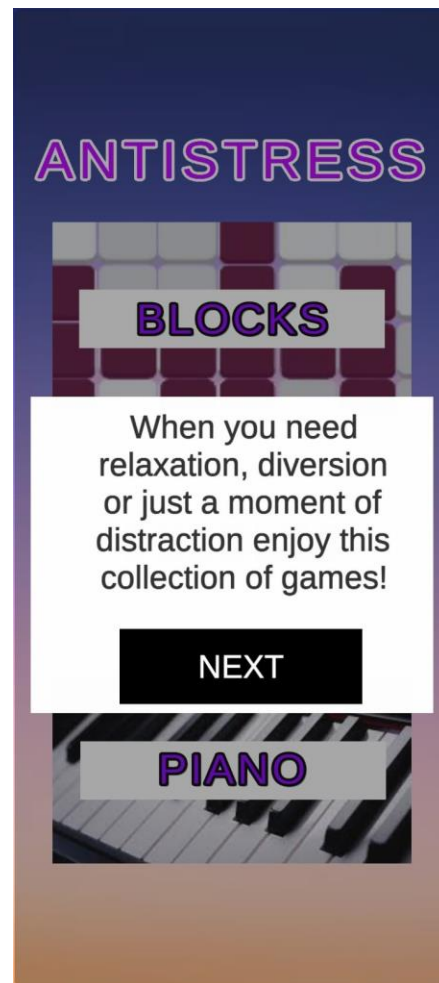


Figure 5.6: Splash Screen 2

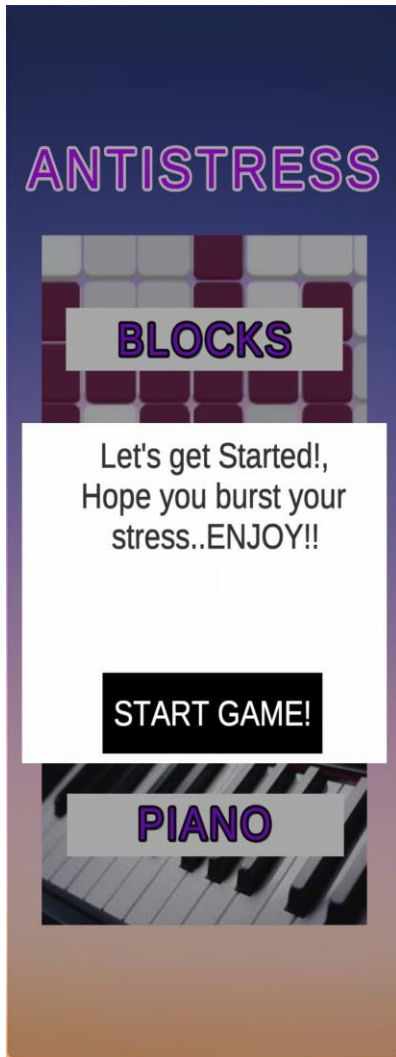


Figure 5.7: Splash Screen 3

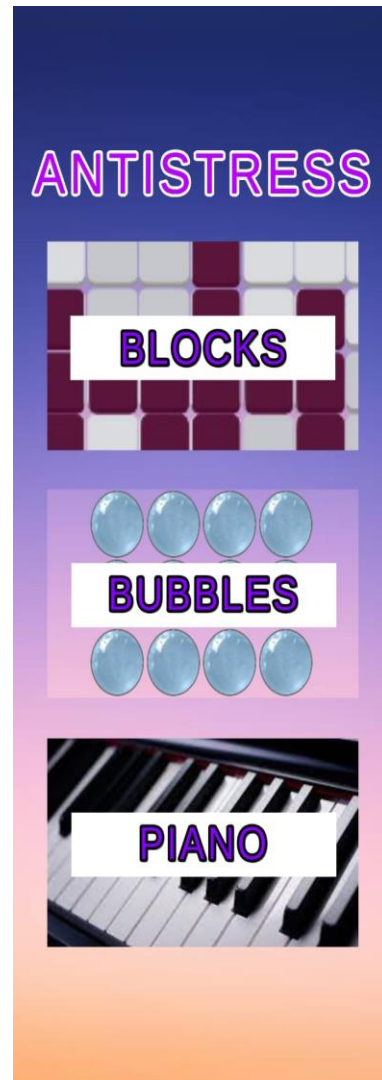


Figure 5.8: Menu page



Figure 5.9: Block puzzle

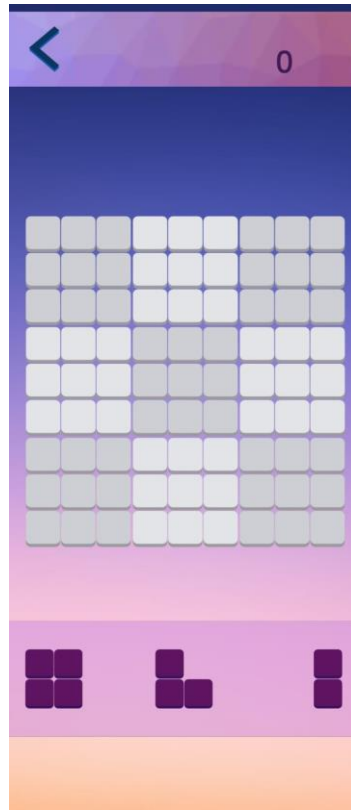
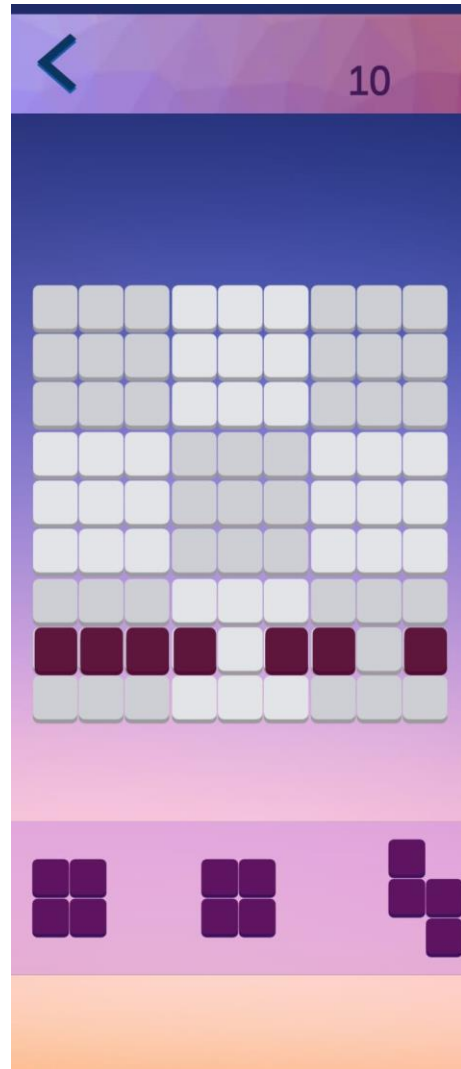
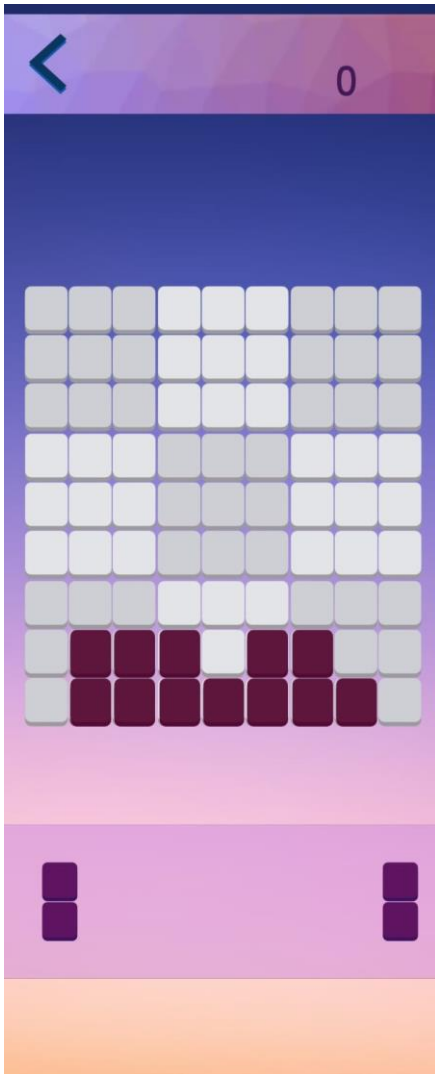


Figure 5.10: Block puzzle game



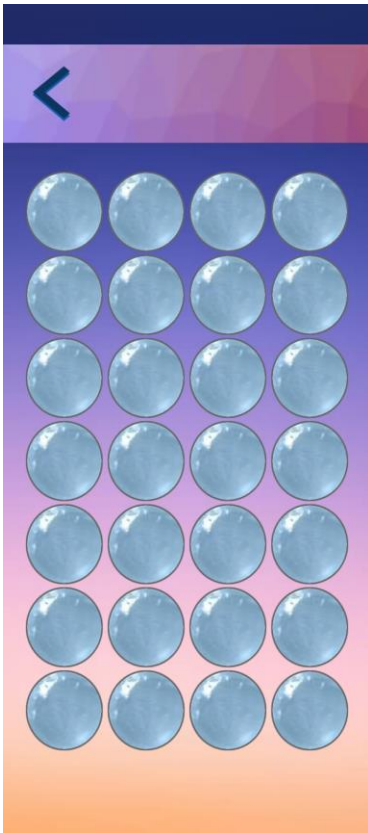


Figure 5.11: Bubble game

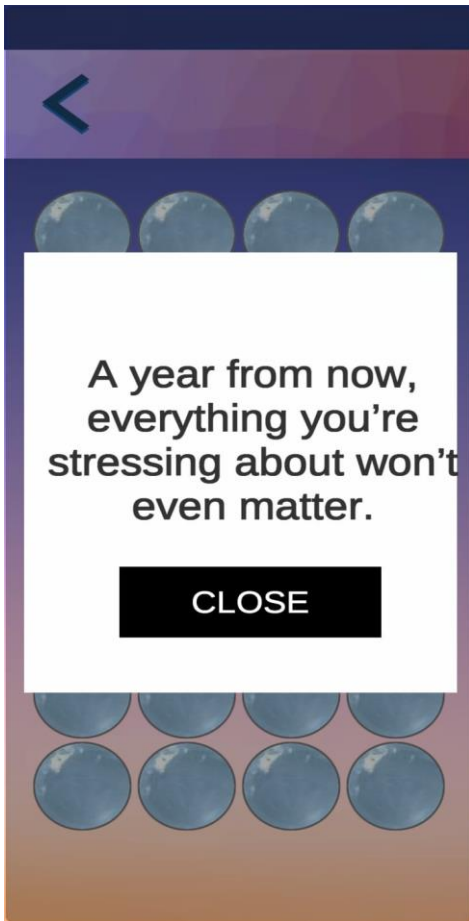


Figure 5.12: Piano Game

6. Future Enhancements

Block

- High score options
- Multiplayer section
- UI changes - Grid's color changing after any particular condition

Bubble

- Scores Window
- Word puzzle can be incorporated into the game replacing the quotes message box
- More interactive using notifications

Piano

- Improve the UI along with actual music notes to follow

Other than these improvements,

A gratitude journal can be added.

Gratitude journals make us pay attention to the good things in life, which we might otherwise take for granted.

As this is an Antistress game, a gratitude journal can be the cherry on the cake.

7. Limitations

- Vibrations can't be felt while popping bubbles, also while placing the bricks.
- Equalizer is missing in the piano game.
- The blocks game doesn't end, although it was on purpose as to avoid the anxiety of game over, it can be seen as a limitation.

9. Conclusion

Using Unity, we have built a game named Anti-Stress. As the name suggests the game helps to relieve stress. The 3 games incorporated in this application are proven to be few of the many stress relieving games. We have tried to merge them together to have a better user experience.

In this digital age, this game can be proven to help reduce stress levels to a greater extent. Not just the games, the UI is designed to soothe eyes and doesn't cause strain. Virtually any game that you find to be truly fun can be beneficial by providing an escape from daily stress, a break from patterns of rumination, or a way to build positive feelings.

10. Bibliography

<https://www.youtube.com/watch?v=HHNxhDN9Cvs>

<https://unity.com/products/unity-teams>

<https://docs.unity3d.com/ScriptReference/SceneManagement.SceneManager.LoadScene.html>

<https://docs.unity3d.com/530/Documentation/ScriptReference/UI.Button-onClick.html>

<https://www.youtube.com/watch?v=ZBxRRAr9hDA>