

**School of Science & Engineering**

**CSC 430101 Introduction to Artificial Intelligence**

**Project # 3 Report**

**Bust the ghost**

**Spring 2023**

Manal Mehdaoui

Taha Chadli

Omolbanin Amiri

Contents

[1. Introduction: 3](#_Toc133666921)

[YouTube link demo: 3](#_Toc133666922)

[Full project Link: 3](#_Toc133666923)

[2. Program Structure: 3](#_Toc133666924)

[3. Probabilistic Inferencing: 7](#_Toc133666925)

[A. Normalization: 8](#_Toc133666926)

[B. Importance: 8](#_Toc133666927)

[4. Conditional Probability Distribution: 9](#_Toc133666928)

[5. Bayesian Inference Algorithm: 9](#_Toc133666929)

[6. How to Simulate a Noisy Sensor: 10](#_Toc133666930)

[7. Determining P(Color/Distance from Ghost): 10](#_Toc133666931)

[8. Updating the Probabilities of All Tiles: 10](#_Toc133666932)

[9. Conclusion 11](#_Toc133666933)

# Introduction:

In this project, we are going to implement a game called "Bust the Ghost" using Unity. The goal of the game is to find the hidden ghost in a 6 x 15 grid by clicking on the cells. Each time the player clicks on a cell, they will receive a color code that indicates the distance between the ghost and the clicked cell. The player will also lose one point from an initial credit for each click, and they can decide to bust a cell if they think the ghost is in that cell. The game ends when the player runs out of credit, or they bust the ghost. To make the game more challenging, the ghost's location is randomly generated based on a prior distribution of the ghost over the location P(Ghost).

To implement the game, we need to use Bayesian inference to update the probability of the ghost's location after each click. We will use a conditional probability distribution P(Color/Distance from Ghost) to decide on the color to display for each click.

## YouTube link demo:

A demo video for the program can be found in the following link:

<https://youtu.be/tcSBHPJhE7w>

## Full project Link:

Link for the project: [CSC 4301- Project3-.zip](https://alakhawayn365-my.sharepoint.com/:u:/g/personal/m_mehdaoui_aui_ma/EX0ALbN1GZFCl6l1yLjpCe0BWiFZLMnRWM-9P7BQCsuk1Q?e=FwoTYt)

# Program Structure:

The program structure of the "Bust the Ghost" game is as follows:

1. Initialize the grid and the probability distribution of the ghost's location.
2. Loop until the game is over:

a. Wait for user input (click on a cell).

b. Simulate a noisy sensor to get the color response of the clicked cell.

c. Determine the conditional probability distribution of the color response given the distance from the ghost.

d. Update the probabilities of all tiles using Bayesian inference.

e. Check if the game is over (either the player has run out of credits or the maximum number of busts has been reached).

3. End the game.

## What is in the game:

**Number of credits** which is 30 at the start and decreases by -1 with each click

**Remaining ghosts:** initially one shows how many ghosts there are.

**Remaining busts:** 10 busts initially and it decreases by -1 for each click in the button “Bust the ghost”

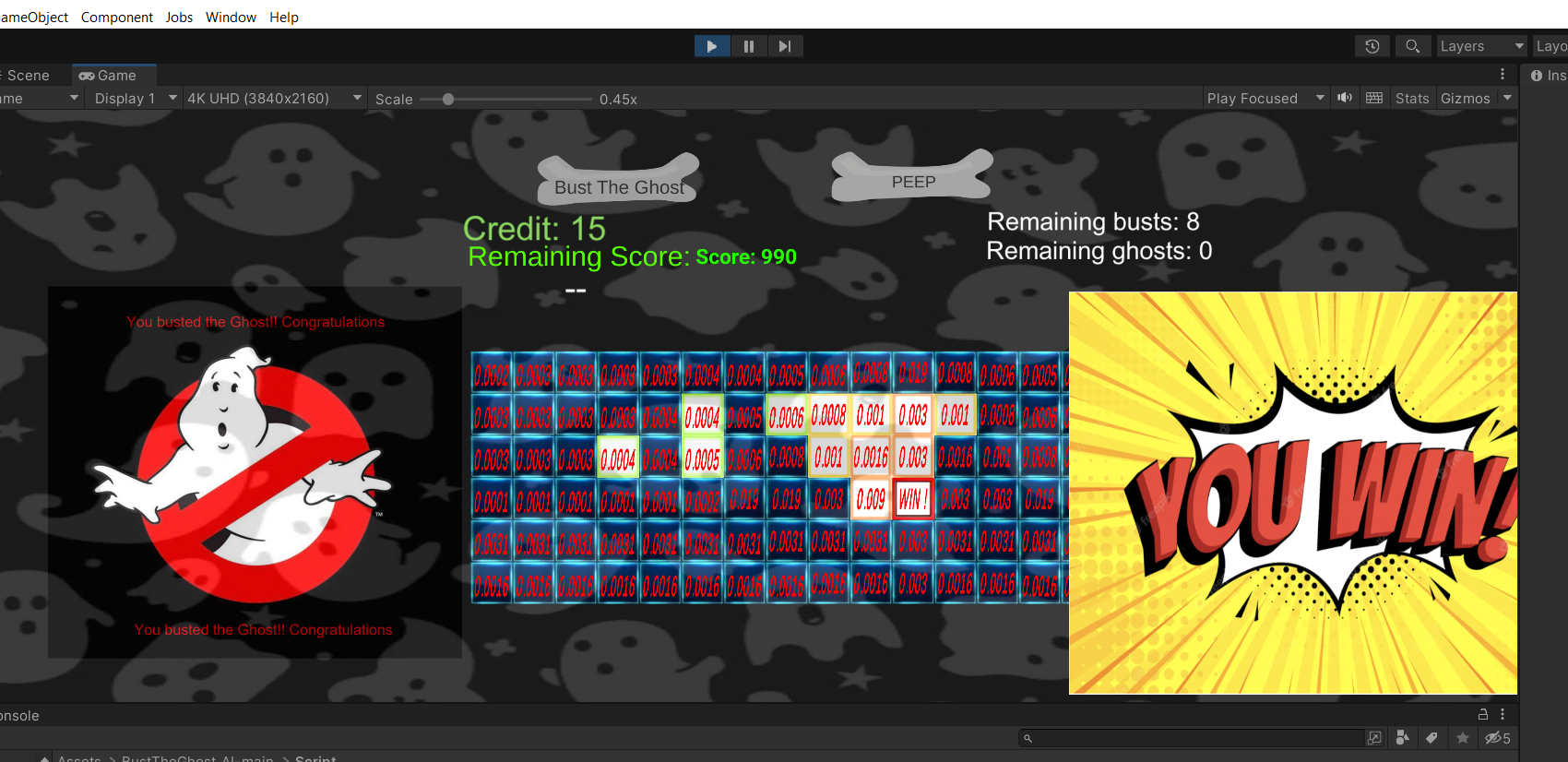
**Remaining score:** -10 if busted wrong cell, +1000 if busted cell with ghost

**Button “Bust the ghost”:** busts the ghost

**The Peep button:** Hides/Unhides the probabilities

**Starting screen**



**If Ghost is busted**

**If number of remaining busts are finished the game is over**

A screenshot of a computer

Description automatically generated with medium confidence

**If number of remaining credits are finished the game is over**

A screenshot of a computer

Description automatically generated with medium confidence

***Clicking on the peep button can hide the probabilities or show them***

In this case we hide them, and the same rules apply

**Finding the ghost**

Graphical user interface, application

Description automatically generated

**NO busts left:**

Graphical user interface, application

Description automatically generated

**No credits left**

**Graphical user interface

Description automatically generated**

# Probabilistic Inferencing:

Probabilistic inferencing plays a crucial role in Bust the Ghost game. The game uses Bayesian inference to update the probability of the ghost's location after each click. The process starts with a uniform prior probability distribution over the possible locations of the ghost on the grid. After each click, the player receives color-coded feedback indicating the distance between the clicked cell and the cell containing the ghost. Based on this information, a conditional probability distribution is used to calculate the probability of the ghost being in each of the possible locations. The prior probability of each possible location is then multiplied by the conditional probability of the color feedback given the distance between the clicked cell and the cell containing the ghost. The resulting product is the unnormalized posterior probability of the ghost being in that location after the current click.

## A. Normalization:

After calculating the unnormalized posterior probability, the probabilities are normalized to ensure they sum up to one. Normalization is crucial to ensure that the updated probability distribution over the possible locations of the ghost remains a probability distribution. The normalization process involves dividing each unnormalized posterior probability by the sum of all unnormalized posterior probabilities over all possible locations of the ghost.

## B. Importance:

Probabilistic inferencing is essential in creating a challenging and engaging game. The use of Bayesian inference and a conditional probability distribution enables the game to become more challenging as the player receives more information about the ghost's location. The player's ability to update the probability distribution over the possible locations of the ghost after each click provides a more informed decision-making process for the player. Additionally, the use of color-coded feedback based on the distance between the clicked cell and the cell containing the ghost, allows players to make more informed decisions, further enhancing the game's challenging nature.

# Conditional Probability Distribution:

To update the probability of the ghost's location after each click, we need to define a conditional probability distribution (CPD) that relates the color response to the distance between the clicked cell and the ghost's location. We assume that the color response depends only on the distance between the clicked cell and the ghost's location. The CPD is defined as follows:

P(Color/Distance from Ghost) = {

0.9 if Distance from Ghost = 0,

0.5 if Distance from Ghost = 1 or 2,

0.1 if Distance from Ghost = 3 or 4,

0.01 if Distance from Ghost >= 5

}

where Distance from Ghost is the number of cells between the clicked cell and the ghost's location.

# Bayesian Inference Algorithm:

The Bayesian inference algorithm used to update the probability of the ghost's location after each click is as follows:

Initialize the probability of the ghost's location as a uniform distribution over all cells.

For each click, do the following:

a. Get the color response of the clicked cell.

b. Update the probability of the ghost's location using Bayes' theorem:

P(Ghostt) = P(Ghostt-1) \* P(Color/Distance from Ghost)

c. Normalize the updated probability distribution:

P(Ghostt) = P(Ghostt) / sum(P(Ghostt))

If the player has run out of credits or the maximum number of busts has been reached, end the game.

# How to Simulate a Noisy Sensor:

To simulate a noisy sensor, we add noise to the distance between the clicked cell and the ghost's location. We assume that the noise follows a normal distribution with a mean of 0 and a standard deviation of 1. We use this noisy distance to determine the color response of the clicked cell.

# Determining P(Color/Distance from Ghost):

We determine the conditional probability distribution of the color response given the distance from the ghost using the following probabilities:

P(Color = Red/Distance from Ghost = 0) = 0.9

P(Color = Orange/Distance from Ghost = 1 or 2) = 0.5

P(Color = Yellow/Distance from Ghost = 3 or 4) = 0.1

P(Color = Green/Distance from Ghost >= 5) = 0.01

We assume that the color response depends only on the distance between the clicked cell and the ghost's location.

# Updating the Probabilities of All Tiles:

To update the probabilities of all tiles after each click, we use Bayesian inference. We update the probability of the ghost's location using the conditional probability distribution of the color response given the distance from the ghost. The algorithm used to update the probabilities of all tiles is as follows:

Initialize the probability of the ghost's location as a uniform distribution over all cells.

For each click, do the following:

a. Get the color response of the clicked cell.

b. Determine the conditional probability distribution of the color response given the distance from the ghost.

c. Update the probability of the ghost's location using Bayes' theorem:

P(Ghostt) = P(Ghostt-1) \* P(Color/Distance from Ghost)

d. Normalize the updated probability distribution:

P(Ghostt) = P(Ghostt) / sum(P(Ghostt))

If the player has run out of credits or the maximum number of busts has been reached, end the game.

# Conclusion

"Bust the Ghost" is a game that showcases how probabilistic inferencing can be used to create an engaging and challenging gaming experience. The game uses Bayesian inference and a conditional probability distribution to make the game more challenging as the player receives more information about the ghost's location. The player's ability to update the probability distribution after each click provides a more informed decision-making process. The game structure involves initializing the grid and probability distribution, simulating a noisy sensor, determining the conditional probability distribution, updating the probabilities using Bayesian inference, and ending the game when the player runs out of credits or reaches the maximum number of busts. Overall, the game is an excellent example of how Bayesian inference and probability theory can be applied in a fun and interactive way.