**COMPUTER VISION**

**CS – GY 6643**

**PROJECT: SNAKE GAME**

**TEAM:**

**HARSH JALUTHARIA – *hj2607***

**MUDIT NIGAM – *mn3439***

**RHUTHVIK DENDUKURI – *rd3377***

Introduction

# Background

The original Snake game, popularized by Nokia's monochrome mobile phones in the late 1990s, is a classic arcade-style video game. The objective is to control a continuously growing snake by guiding it to eat food items while avoiding collisions with its own body or the boundaries. As the snake grows longer with each piece of food consumed, the challenge lies in navigating through an increasingly crowded and confined space.

The game's simplicity and portability contributed to its enduring popularity, making it an ideal testbed for computer vision projects focused on tracking movement, detecting collisions, and developing advanced features like obstacle avoidance or path planning algorithms.

# Motivation

A great chance to create and test computer vision algorithms in a controlled setting is the Snake game. It presents various non-trivial issues for computer vision systems despite having straightforward gameplay mechanisms. Robust vision algorithms are needed to identify food items, follow the snake's movement accurately, and detect impacts with objects or the snake's own body. Furthermore, the continuous nature of the game and its rising complexity as the snake lengthens can be utilized to assess how well vision techniques operate and how scalable they are under different circumstances. Starting with this well-defined and well-known game allows researchers to concentrate on creating unique vision techniques without being distracted by intricate rendering or physics models.

# Objectives

## Implement a Snake Game

The primary objective is to develop the core mechanics of the snake game, encompassing essential elements such as snake movement, food and hurdle generation and a scoring system. By recreating the classic gameplay, we aim to provide a familiar yet enjoyable experience for players.

## Integrate Computer Vision

Leveraging computer vision algorithms, we seek to detect objects within the game environment and enable players to control the snake’s movements using real-world objects captured by the system’s web camera. This integration adds a unique layer of interactivity to the game, allowing players to engage with the virtual world in innovative ways.

## Enhance Gameplay Experience

The goal is to create an engaging and intuitive gameplay experience by seamlessly blending computer vision technology with the classic game mechanics.

## Scalability and Performance

It is crucial to design the system to be scalable and efficient, capable of running in real-time in standard hardware while delivering a smooth experience. By optimizing algorithms and leveraging hardware acceleration where possible, we aim to ensure that the game performs reliably across a range of devices and environments.

Project Overview

# Description

This project reimagines the Snake Game by integrating computer vision. Players control the snake, collecting food and avoiding obstacles. What sets it apart is the ability to control the snake using real-world objects detected through a camera feed. Computer vision algorithms analyze the feed in real-time, translating object positions into commands for the snake's movement. This immersive gameplay experience blends reality with virtuality, offering dynamic interaction. Whether navigating household items or custom obstacles, players enjoy a fusion of nostalgia and innovation.

# Technologies Used

1. Python
2. OpenCV
3. Numpy
4. Tkinter
5. Random Module

# Features

* Real - Time Object Detection: The game utilizes CV algorithms to detect real – world objects, allowing players to controls the snake’s movements using their surroundings.
* Dynamic Hurdle generation: Hurdles are dynamically generated within the game frame, providing challenges for players to navigater around.
* Restriction – less movement: Unlike the classic game, the movement of the snake is not restricted to the 4 directional movements. It is made to be a free flow all around the game frame.