Doom/portal Style game:

Combining the fast-paced action of *Doom* with the mind-bending portal mechanics of *Portal*. Players will find themselves in a nightmarish, labyrinthine world filled with relentless demons.

**Stakeholders:**

The project involves several key stakeholders:

* **Game Developers**: Responsible for game design, coding, and overall development.
* **Game Players**: The primary audience who will play and enjoy the game.

**Potential Research Avenues:**

**AI Behavior:** Developing advanced AI algorithms for enemy movements and attack patterns is essential to create challenging and engaging gameplay. The AI must adapt to the unique portal mechanics and player strategies.

**Portal Physics:** Understanding and implementing portal physics is a critical aspect of the game's mechanics. Research into how to render scenes through portals while maintaining gameplay fluidity will be necessary.

**Data Processing Needed:**

The game will require data processing for various tasks:

**Real-time Physics Simulation:** Accurate physics simulations are essential for rendering portal mechanics realistically. This involves handling collisions, physics interactions, and portal-related effects.

**Procedural Level Generation:** To provide players with diverse and dynamic gameplay experiences, procedural level generation algorithms must be developed. These algorithms will create labyrinthine levels filled with demons and portals.

**Enemy Behavior:** Implementing AI routines for demon characters, including their movements, reactions, and combat strategies, is crucial. The enemy must respond intelligently to the player's actions and the presence of portals.

**Rendering Techniques:** Rendering the first-person perspective, possibly through techniques like ray casting or binary space partitioning, is necessary to create a visually immersive experience.

**Current Problem/Initial Ideas for a Solution:**

One significant challenge is seamlessly integrating portal mechanics into the frantic gameplay of *Doom*.

**Programming Language(s) to be Used:**

The game can be developed using either Java or C# within the Unity game engine. Unity offers 2D game development capabilities and a robust set of tools to streamline the development process. Developing a game engine from scratch in Java using LWJGL 3, although possible, would be a significantly more complex and time-consuming endeavour.

**Ideas for a Suitable GUI:**

The game's GUI will be designed for an immersive and seamless gaming experience, featuring:

* **Health and Ammo Display**: Clear and intuitive indicators for player status.
* **Display weapon**: Display First person perspective of user holding a weapon.

Top-down shooter in randomly generated maze:

Developing a top-down shooter set in randomly generated mazes offers an engaging fusion of precision movement, intense combat, puzzle-solving, and exploration. Here, we delve into key aspects of this game concept:

**Stakeholders:** The project's key stakeholders encompass:

1. **Game Developers:** These individuals or teams are central to the game's design, coding, and overall development.
2. **Game Players:** The primary audience is enthusiastic gamers seeking a challenging and dynamic gameplay experience.

**Potential Research Avenues:** Crucial research areas include:

1. **Random Maze Generation:** Exploring algorithms to generate mazes that are both challenging and fair, ensuring each playthrough offers a fresh and enjoyable experience.
2. **AI Behavior:** Developing sophisticated AI algorithms for enemy movements and attack patterns to create engaging encounters.
3. **Control Schemes:** Investigating optimal control schemes that enable precise player movement within the maze, a key element of the game's challenge.

**Data Processing Needed:** To realize the game's vision, the following data processing tasks are imperative:

1. **Maze Generation:** Generating randomly generated mazes that strike the right balance between complexity and playability.
2. **Enemy AI:** Processing enemy behaviors, pathfinding, and interactions with the player to provide challenging and dynamic gameplay.
3. **Physics Simulation:** Handling player movement and collision detection within the maze to ensure responsive and immersive controls.

**Current Problem/Initial Ideas for a Solution:** The main challenge lies in seamlessly combining precision movement with intense combat in the context of randomly generated mazes. A potential solution involves meticulous level design to maintain a balance between navigational complexity and exciting battles. Careful tuning of enemy AI and control schemes will also be essential.

**Programming Language(s) to be Used:** The game can be developed using either Java or C# within the Unity game engine. Unity offers 2D game development capabilities and a robust set of tools to streamline the development process. Developing a game engine from scratch in Java using LWJGL 3, although possible, would be a significantly more complex and time-consuming endeavour.

**Ideas for a Suitable GUI:** The graphical user interface (GUI) should enhance the gaming experience and provide essential information to players. Key GUI elements encompass:

1. **Health and Healing Section:** Displaying health status and an option for manual healing to help players survive the challenges.
2. **Weapon Reload Section:** Showing the current weapon and ammunition status, along with a manual reload option for strategic gameplay.

2D Call of duty Zombies clone:

**Stakeholders:** The development of a 2D Call of Duty Zombies clone involves two primary stakeholders:

1. **Game Developers:** These individuals or teams will be responsible for the game's design, coding, and overall development. They are critical in bringing the concept to life and ensuring an engaging zombie-slaying experience.
2. **Game Players:** The primary audience for the game consists of gamers who enjoy survival horror and wave-based gameplay. Their feedback and enjoyment are paramount for the project's success.

**Potential Research Avenues:** Several research areas are critical for the development of a successful 2D Call of Duty Zombies clone:

1. **Enemy Behaviour and AI:** Realistic zombie AI and movement patterns must be developed to create an immersive and challenging experience. Zombies should exhibit lifelike behaviours such as pursuing players and responding to environmental cues.
2. **Balancing and Progression:** Ensuring that the game's difficulty curve and player progression are finely tuned is essential. The gameplay should become increasingly challenging without becoming overly punishing, maintaining player engagement.

**Data Processing Needed:** To achieve the game's goals, various data processing tasks are required:

1. **Physics Simulation:** Accurate physics simulations are needed for projectile trajectories, explosions, and zombie movements. This ensures that combat feels realistic and satisfying.
2. **Enemy AI:** Processing zombie behaviours, pathfinding, and interactions with the environment is crucial for creating challenging and dynamic encounters.

**Programming Language(s) to be Used:** For the development of this 2D game, either Java or C# can be employed, utilizing the Unity game engine. Unity provides a robust platform for creating 2D games with complex mechanics and physics. Developing a game engine from scratch in Java using LWJGL 3, although possible, would be a significantly more complex and time-consuming endeavour.

**Graphical User Interface (GUI):** The game's graphical user interface (GUI) should provide essential information to players while maintaining the immersive atmosphere. Key GUI elements include:

1. **Round Counter:** Displaying the current round or wave to keep players informed about their progress.
2. **Weapon and Ammo:** Clear indicators for the selected weapon and available ammunition to aid players in decision-making during intense battles.
3. **Points and Interactable Items:** Showing the player's points or currency, as well as highlighting interactable items or power-ups to enhance gameplay strategy.

Overall, the GUI should be intuitive, seamlessly integrated into the game's visual style, and contribute to the immersive and pulse-pounding zombie-slaying action that defines the game.