**STE Mission Planning Toolkit - Presentation Document**

**Overview**

The **STE Mission Planning Toolkit** is a Python-based GUI application developed using **Tkinter** that provides an intuitive menu-driven system for launching various simulation tools, scenarios, terrain configurations, and tutorials. The primary objectives of this toolkit are to **streamline the workflow**, **ensure easy access** to required applications, and **enhance user experience** through a structured and well-organized interface.

**How It Works**

The application consists of a **main menu** and **submenus**, allowing users to navigate through different functionalities effectively. The key components include:

**1. Main Menu**

* The **main menu** is a **central hub** that provides access to the following categories:
  + **Tools** (Simulation applications like VBS4 and DXTRS)
  + **Scenarios** (Different mission scenarios to launch)
  + **Terrain** (Loading different training environments)
  + **Help - Tutorials** (Guides for mission building and terrain loading)
  + **Exit Button** to close the application
* The menu is **locked at 1600x800 resolution**, ensuring a consistent UI across all screens.
* A **background image** is applied dynamically to maintain a professional appearance.

**2. Tools Menu**

* The **Tools menu** provides buttons to launch different simulation tools:
  + **Launch 3D Wargame (VBS4)** – Runs VBS4 using a batch script.
  + **VBS4 Setup** – Directly opens the VBS4 Launcher (VBSLauncher.exe).
  + **Launch BVI** – Runs the BVI software using a batch file.
  + **Launch DXTRS** – Executes the DXTRS tool.
* Each button runs a specific **batch file or executable**, ensuring that tools are launched efficiently.

**3. Scenarios Menu**

* The **Scenarios menu** provides buttons for launching predefined mission scenarios.
* Clicking on any scenario button triggers a message box confirming the launch.

**4. Terrain Menu**

* The **Terrain menu** allows users to load different training environments.
* Each button represents a **training location** such as **NTC, Stewart, Muscatuck, or Orlando**.
* Selecting a terrain triggers a message box indicating that the terrain is being loaded.

**5. Help - Tutorials Menu**

* This menu provides access to guides and tutorials:
  + **Build a Scenario** – Provides information on scenario creation.
  + **Load a Terrain** – Guides users on how to load and modify terrain configurations.

**Technical Implementation**

**1. Tkinter GUI Setup**

* The GUI is built using **Tkinter**, Python’s built-in library for creating graphical user interfaces.
* The **main window (root) is initialized with a fixed size (1600x800)** to prevent resizing.
* **A background image is loaded and resized dynamically** to fit the window properly using **Pillow (PIL)**.

**2. Batch File & Executable Launching**

* The toolkit uses **subprocess.Popen()** to launch external batch files and executables.
* The **launch\_application() function** is responsible for executing these files while handling errors gracefully.

**Submenu Navigation**

* Each main menu option opens a **submenu window** using **Tkinter’s Toplevel() widget**.
* Submenus **inherit the background image** and maintain the same visual style.
* Buttons are **organized in a structured manner with even spacing** for consistency.
* A **"Back" button allows users to return to the main menu**, while an **"Exit" button closes the entire application**.