1. Game Concept:
   * Idea:

In this game we have a mission to rescue our partner from the clutches of the enemy. We will face two thrilling stages of challenges

Stage 1:

Prepare for intense combat as we confront the enemy with our trusty default weapon, a sharp knife. Defeat them and earn valuable coins that will be our ticket to weapon upgrades.

Stage 2:

Equip ourselves with our upgraded arsenal and confront the formidable boss. We will need to fire five precise shots to bring them down. Upon defeating the boss, seize the key to our partner's prison and free them, clearing Stage 2.

* + Genres
    1. Action
    2. Shooting
    3. Adventure
    4. Rescue
    5. Combat
    6. Boss Battle
    7. Heroic
  + Targeted Audiences:

Age 12-30

1. Choice of Game Engine

The “Unity” is the game engine for the development of this game because of its characteristics as mentioned below:

* Unity's Cross-Platform Capabilities
* Vast Asset Store and Ecosystem
* User-Friendly Development Environment
* Robust 2D and 3D Capabilities
* Strong Community and Support



1. Game Mechanics:
   * Control:
     1. Basics
        + Movement-> W, A, S, D keys or Arrow keys
        + Jumps -> spacebar
        + Hide-> B
        + Attack -> Left Mouse Button or L
     2. Special Abilities
        + Change weapons -> Q
   * Camera And Views
     1. Mouse Movement
     2. Zoom in/out -> scroll mouse
   * Menu and UI
     1. Pause-> Esc
     2. Inventory -> I
   * Rules
     1. *Objective:*

Free the partner character from the enemy's cage.

* + 1. *Character Health:*

The hero character has a health bar.

* + 1. *Enemy Interaction:*

Players must defeat or bypass enemy characters to progress.

* + 1. *Items and Inventory:*

Players can collect items such as health potions, keys, and power-ups that are stored in the hero's inventory.

* + 1. *Level Completion:*

Each level is considered complete when the partner character is successfully freed.

* + 1. *Level Progression:*

Players advance to the next level after completing the objectives of the current level.

vii. Player Defeat:

If the hero character's health is depleted or the partner character is not freed within the level's time limit, the game ends in failure.

viii. Scoring:

Players may earn points for defeating enemies or collecting items.

ix. Character Abilities:

Describe the special abilities and skills that the hero character possesses, such as attack moves, defensive abilities, or special powers.

x. Difficulty Levels:

The game may offer multiple difficulty levels, affecting factors like enemy strength, and time limits.

xi. Game Controls:

All the game controls as mentioned above.

1. High-level Diagram:
2. Arts and Visualization:
   * Environment
     1. Stage 1:

* Terrain
* Mountains
* Grass
* Sky
* Houses
* Offices
* Etc.
  + 1. Stage 2:
* Terrain
* Mountains
* Grass
* Sky
* Houses
* Offices
* Etc.
  + 1. Visual References:
* Free Online Asset Repositories (Sketchfab, TurboSquid, and Mixamo)
* Unity Asset Store
  + Characters Designs:
    1. Stage 1:
    - Playable character
    - enemies
    1. Stage 2:
    - Boss
    - Playable character
    - Enemies
    1. Visual References:
       - Free Online Asset Repositories (Sketchfab, TurboSquid, and Mixamo)
       - Unity Asset Store
  + Collectable Obstacles:
    1. Stage 1:
       - Coins
       - Power boots up supplement
       - Map of stage 2
       - Weapons of the enemies
    2. Stage 2:
       - Coins
       - Key
       - Map of the prison
       - Bullets
    3. Visual References:
       - Free Online Asset Repositories (Sketchfab, TurboSquid, and Mixamo)
       - Unity Asset Store
  + Weapons:
    1. Basic:
       - Sharp knife
    2. Upgraded:
       - Pistol
       - Shotgun
       - Rifle
       - Revolver
  + Visual References:
    1. Free Online Asset Repositories (Sketchfab, TurboSquid, and Mixamo)
    2. Unity Asset Store

1. Audio and Music:

* Footsteps:
  + When our playable character is walking
* Shooting sound:
  + When both our character and the enemies will fire.
* Knife attack:
  + By default, we have a knife that emits a sound on attack
* Dropping items:
  + When we drop any item from our bag. And the covers of bullets after fire
* Picking up items:
  + When we pick any obstacle in the game.
* Changing weapons:
  + When we change the weapon, it will be done with a sound.
* Lighting, thunder and rains:
  + The rain and the thunder will make seamless noise.
* Enemies noise:
  + When they will walk and shout when they will see us.
* Pausing music:
  + When we pause the game, it will play a music.
* Waiting music:
  + When we will do nothing, it will also play a music.
* Stage changing music:
  + When we go from stage 01 to stage 02.
* Weapon upgradation music:
  + When we successfully upgraded the weapons, it will make hurray music.

1. Level Design:
   * Stage 01:
     1. Objective:

Our motive to gain the coins and clear this stage by killing the enemies using a sharp knife, and gain the more and more coins as much as possible to upgrade the weapon enough.

* + 1. Overview:

Enemies are walking here and there and observing that no one can interrupt them during their operation. We will find them and kill through back of them.

* + 1. Layouts:

There is forest with some trees, houses and offices and some other obstacle (3D objects) that are displayed in the game scene.

* + 1. Challenges:

Our first challenge in the first stage is to clear stage 01 by killing the enemies and collect the coins to upgrade the weapons before going to the next stage.

* + 1. Characters included:
* Playable character (player)
* Enemies
* Our partner under control of the enemy
  + Stage 02:
    1. Objective:

Kill the boss and get the key of the prison of our partner and free him.

* + 1. Overview:

We will attack the boss and kill him by using the upgraded weapon because the power of the boss is higher than the normal enemies (in stage 01). Boss will also attack us and try to kill us and also defend himself when we attack him by using AI.

* + 1. Layouts:

The sample scene as the stage but it contains the boss that will be 10times more powerful than the previous enemies

* + 1. Challenges:

boss get the key and free our partner to end the game.

* + 1. Characters included:
* Boss
* Partner
* Character

1. Technical Requirements:

* Platform Compatibility:

The game will be compatible with Windows and Android,

* Minimum Hardware Requirements:

We are required a laptop with minimum specification of 8gb ram and 128GB SSD hard drive



* Graphics and Display:



1GB graphic card is not necessary but enough for the development

* Input Devices:

Keyboard and mouse



* Performance Optimization:
* Implement efficient coding practices to ensure smooth gameplay and minimal resource consumption.
* Load and unload assets dynamically to optimize memory usage.
* Programming Language:
* C# sharp is the must required for the game development as a backbone to make logics and all functionalities



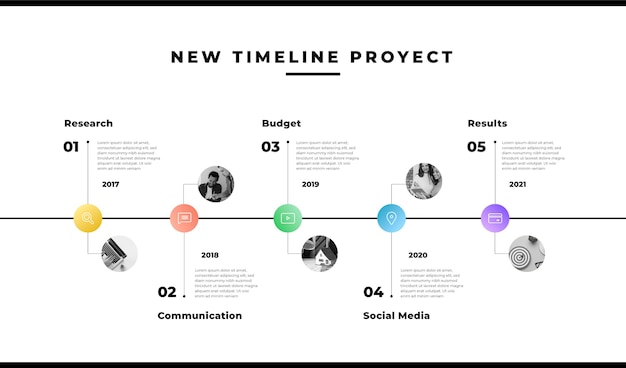
* Compatibility and Version Control:
* Git as version control system to track changes and facilitate collaborative development.



1. Development Time Line:

This project is required about 2 months almost equal to 8 weeks. The strategy for the game development within the time line is mentioned below:

* Week 1: Concept and Design
* Define the game concept, genre, and core gameplay mechanics.
* Create a design document, including game mechanics, level layouts, and art style.
* Begin asset creation or gather pre-existing assets if available.
* Initial design and concept review.
* Week 2: Prototyping
* Create a basic prototype with essential gameplay elements to test your ideas.
* Week 3: Development
* Start building the game's core functionality, including player controls, game mechanics, and basic levels.
* Week 4: Art and Assets
* Continue creating or collecting assets, including 2D/3D models, textures, and sound effects.
* Week 5(Level Design):
* Design and create additional levels, balancing gameplay and difficulty.
* Week 6: Refinement and Bug Fixing
  + Polish the game, fix bugs, optimize performance, and improve user experience.
* Week 7: Playtesting and Finalization
  + Conduct playtesting to make necessary adjustments.
  + Finalize the game, perform rigorous testing, and prepare for release.
* Week 8: Submission
  + If applicable, prepare for and submit the game to relevant platforms (e.g., Steam, App Store).



1. 3D:

1. Character Models:

- Hero character model.

- Enemy character models (various types for diversity).

- Boss character model for the final stage.

- Partner character model (to be rescued).

2. Environment:

- Cage model for the partner.

- Cage key model.

- Coins or gems for collecting as currency.

- Knife model for the basic weapon.

- Upgraded weapons, such as swords, guns, or magic staffs.

- Various props like crates, barrels, and barricades to make the environment more interactive.



3. Animations:

- Animation packs for character movement, attacks, and enemy AI.

- Animation for opening the cage.

4. Audio:

- Background music and sound effects for combat, collecting coins, and other interactions.

5. Particle Effects:

- Visual effects for attacks, hits, and weapon upgrades.

6. UI Elements:

- Health bars for characters and enemies.

- Coin counter and inventory for displaying collected coins and equipped weapons.

- Stage transition screens.



7. Scripts and Plugins:

- Unity Asset Store offers many scripts and plugins for character control, enemy AI, and game mechanics like coin collection and level progression.

8. Terrain and Level Design:

- Terrain assets for creating different stages.

- Decorative elements like trees, rocks, and buildings.

1. Scene Management:

* Start Scene:
* This is the initial scene that players see when they start your game.
* In this scene, you can have a main menu, options, and settings.
* Implement a "Play" button that loads the first level.



* Game Levels:
* Each game level is a separate Unity scene.
* Separate scene for the first stage where the hero battles enemies with a basic knife.
* When the player completes the first stage (collects enough coins or defeats all enemies), use Unity's scene management to load the next stage.
* Boss Fight:
* For the final stage, there is a separate scene where the hero confronts the boss character.
* When the boss is defeated, next scene is loaded.
* End Scene:
* After defeating the boss and rescuing the partner, there is a scene that displays the victory message.
* From here, we can return to the main menu or offer the player to start a new game.



Methods for scene managements:

* SceneManager.LoadScene()
* SceneManager.LoadSceneAsync()
* SceneManager.UnloadScene() etc.

1. Physical parameters:

In Rescue Raider game development context, we may need to consider various physical parameters

* Physics Engine:
* Rigid body
* Collider
* Gravity
* Friction and Bounciness:
* Physics Materials:
* Joint Constraints:
* Character Controller:
* NavMesh
* Particle System:
* Lighting and Shading:
* Audio
* Character Movement
* Ragdoll Physics
* Time Scale:

1. Development Requirements

* Hardware Requirements
  + SSD hard drive with minimum storage or HDD with at least 258gb
  + RAM with minimum 4GB
  + Graphics with 1GB or not required
  + CPU with i3 3rd generation
  + Operating System should be any window or Linux or any other with 64\* bites
  + Input devices (mouse and keyboard) are mandatory
  + Mobile Device Specifications minimum 4GB RAM and 128GB hard internal storage



* Software Requirements
  + Operating System: window or Linux
  + Graphics Drivers: up to date
  + Software Updates: latest version of the software
  + Mobile specification: Android versions



1. Algorithmic Reasoning

* For our Player:
  + Movement of the characters
  + Pick up the weapon
  + Sound on picking up the weapons
  + Sound on changing the weapons
  + Sound when player will walk
  + Different sounds on different floor (on grass, on road, on wood floor)
  + Sound when shoot the gun
  + Pick up the obstacles (coins, power boost ups, lives, key etc.)
  + Health bar will increase or decrease
  + Lives will reduce when he will be killed
  + Open the cage of the partner
  + Gain the power by picking the power boost up obstacles
  + Jump and hide and can shoot the gun
* For enemies:
  + Will be killed when they will shoot by player
  + They will follow and attack the player when the player reaches in the specific region (radius) of the enemies
  + Walk here and there by using the loop
  + They will be replaced with coins when they will be killed
  + Their gun will defect the player 5 time less than the player gun
* Boss with AI:
  + 5 times powerful than the other enemies
  + He will also have the health bar
* Partner and cage:
  + Initially cage is locked
  + Cage will be open after killing the boss
  + Cage will be opened by using the key
  + Cage gate will be opened with animation
  + And partner will come out only when gate is opened
  + Game will be end when the partner is out of cage
* Other Scripting:
  + Coins will be added to the player bank when he reached at the same position of the coins
  + Coins will be disappeared when they are added to the player’s bank
  + Key will be show on the screen when key is collected
  + Gate will be automatically when reached at cage gate

