3.2.1 Organizational Requirements

3.2.1.1 Availability

Availability requirements for a platformer game are related to ensuring that the game is available and accessible to players at all times. Here are some availability requirements to consider**:**

- **Uptime:** Ensure that the game is available to players for the majority of the time. This means minimizing server downtime and addressing any technical issues that could impact the game's availability.
- **Accessibility:** Make the game accessible to a wide range of players, including those with disabilities. Ensure that the game is compatible with assistive technologies and that it meets accessibility guidelines.
- **Responsiveness:** Ensure that the game responds quickly to player input and provides a smooth and lag-free gameplay experience. This means optimizing game code, minimizing network latency, and using efficient server infrastructure.
- **Scalability:** Design the game architecture to be scalable, so that it can handle increases in player traffic and usage without impacting availability or performance.
- **Disaster Recovery:** Have a disaster recovery plan in place in case of unexpected events that could impact the availability of the game, such as server crashes or natural disasters.

3.2.1.2 Latency

Latency requirements for a platformer game are related to ensuring that the game responds quickly and smoothly to player inputs, providing a seamless and enjoyable gameplay experience. Here are some latency requirements to consider:

- **Input Responsiveness:** Ensure that the game responds quickly and accurately to the player's input, with minimal delay or lag. This is particularly important for a platformer game where precise timing and movement are crucial.
- **Game Engine Optimization:** Optimize the game engine to reduce processing time and minimize input lag. This can be achieved by using efficient algorithms, reducing unnecessary computations, and optimizing the game engine's architecture.
- **Hardware Optimization:** Optimize the game for different hardware configurations to minimize input lag. This can be achieved by optimizing the game engine for different CPU and GPU architectures, and testing the game on a wide range of hardware configurations.
- **Player Feedback:** Provide immediate and clear feedback to the player when they perform an action. This can include visual and audio cues, such as animation or sound effects, to indicate that the game has registered the player's input.

3.2.1.3 Monitoring

Monitoring requirements for a platformer game project are related to the measurement and management of various aspects of the game during development and after release. Here are some monitoring requirements to consider:

- **Performance Monitoring:** Monitor the performance of the game, including frame

rate, loading times, and resource utilization, to identify and diagnose performance issues.

- **Bug Monitoring:** Implement a bug tracking system to monitor and track reported bugs and issues. This can help prioritize and manage bug fixes.
- **User Behavior Monitoring:** Monitor user behavior, including playtime, progression, and retention, to understand how users interact with the game and identify areas for improvement.
- **Server Monitoring:** Monitor game servers to ensure that they are running smoothly and efficiently, and to identify and diagnose issues that may impact gameplay.
- **Security Monitoring:** Implement security monitoring to detect and prevent cheating, hacking, and other security issues that may impact the integrity of the game.
- **Analytics:** Implement analytics tools to measure and analyze user behavior, engagement, and other metrics that can inform game development and optimization.

3.2.1.4 Maintenance

Game maintenance is an important aspect of game development that ensures the game remains functional and enjoyable for players after its release. It involves ongoing updates and fixes to address bugs, glitches, and other issues that may arise.
For this platformer-style game, maintenance will be crucial in ensuring the game remains playable and enjoyable for players. This includes:

- **Bug fixes:** Game developers must monitor the game closely to identify and fix any bugs that may arise. These can include issues with player movement, collisions, enemy behavior, and other gameplay elements.
- **Balancing updates:** As players become more experienced with the game, they may identify areas where the difficulty needs to be adjusted or certain elements need to be rebalanced. Developers will need to monitor player feedback and make necessary adjustments to ensure the game is challenging, but not frustrating.
- **Content updates:** To keep players engaged and interested in the game, new content such as levels, enemies, and power-ups can be added over time. This can help to keep the game fresh and encourage players to continue playing.
- **Performance optimization:** As new hardware and operating systems are released, game developers must ensure that the game remains compatible and runs smoothly on these new platforms. This may require updates to the game's code or optimizations to improve performance.

3.2.1.5 Operations

- **Network and server infrastructure:** Games that involve multiplayer gameplay typically require a robust network and server infrastructure to handle the traffic and data generated by users. Developers need to ensure that the game's backend systems are designed to be scalable, redundant, and secure. This means that they need to consider things like load balancing, failover mechanisms, network topology, and security protocols when designing the infrastructure. They also need to ensure that the infrastructure can handle high volumes of traffic and that it is able to provide a consistent user experience across different devices and platforms.
- **Game deployment and management:** Developers need to ensure that the game is deployed and managed in a way that ensures maximum uptime and availability.

This means that they need to consider things like continuous integration and deployment, monitoring and alerting, and incident response procedures. They also need to ensure that they have appropriate backup and disaster recovery plans in place to minimize the impact of any disruptions or outages.

- **Player support and engagement:** To ensure a positive player experience, developers need to provide appropriate support and engagement mechanisms. This includes things like providing clear and comprehensive documentation, offering responsive customer support, and engaging with players through social media and other channels. They also need to ensure that they have appropriate mechanisms in place to address player feedback and concerns.

3.2.1.6 Standards Compliance

- **Payment processing:** Many games rely on in-app purchases and microtransactions to generate revenue. To ensure compliance with relevant regulations and standards, developers must implement secure payment processing systems that protect user data and adhere to industry standards such as the Payment Card Industry Data Security Standard (PCI DSS). This includes things like encrypting sensitive data, using secure payment gateways, and implementing fraud detection mechanisms.
- **Data privacy:** As games collect and process user data, developers must ensure that they comply with relevant data privacy regulations such as the General Data Protection Regulation (GDPR) and the California Consumer Privacy Act (CCPA). This means that they need to implement appropriate data protection measures, such as encryption, access controls, and data retention policies. They also need to provide users with clear and transparent information about what data they collect and how they use it, and obtain user consent where necessary.
- **Accessibility:** Games should be designed to be accessible to all users, including those with disabilities. Developers need to ensure that they comply with accessibility standards such as the Web Content Accessibility Guidelines (WCAG) and the Americans with Disabilities Act (ADA). This means that they need to consider things like color contrast, font size, and support for assistive technologies such as screen readers and keyboard navigation.
- **Content moderation:** As games allow users to interact with each other, developers must ensure that they comply with relevant content moderation regulations and standards. This includes things like implementing reporting mechanisms, establishing community guidelines, and enforcing appropriate sanctions for violations of those guidelines. They also need to ensure that they comply with relevant laws and regulations regarding hate speech, harassment, and other harmful behaviors.

3.2.1.7 Portability

Game portability refers to the ability of the game to be played on different platforms and operating systems without any significant changes to the codebase. This is an important aspect of game development, as it allows the game to reach a wider audience and ensures that players can enjoy the game regardless of the device or system they are using.

For this platformer-style game, we aim to ensure high portability across multiple platforms, including desktops, laptops, and mobile devices running on Windows, macOS, and Linux

operating systems.
To achieve this, we will use game development frameworks and engines such as Unity or Unreal Engine, which provide cross-platform support and allow developers to write code once and deploy to multiple platforms with minimal effort. We will also ensure that the game's assets (such as images, sound effects, and music) are optimized for different resolutions and aspect ratios to ensure the game looks and feels consistent across different platforms.
Moreover, we will make sure that the game's user interface is designed with portability in mind. This means that the game's menus, buttons, and other user interface elements will be designed to work well on different screen sizes and resolutions. Additionally, we will ensure that the game's control scheme is designed to work well with different input devices, such as keyboards, mice, gamepads, and touchscreens. By prioritizing portability, we aim to make the game accessible to as many players as possible, regardless of their device or operating system.