

# Stolen Treasures: GDD

## Team Roktleeg

Kushagra Kharbanda(2020101002)

Harshita Gupta(2020101078)

## Game Overview

Stolen Treasures is a 2D platformer with puzzle-solving elements that follows the adventures of Arjun, a charming thief with a Robin Hood-like motive. The game explores themes of morality, happiness, and self-discovery as Arjun embarks on a quest for treasures, powers, and ultimately, understanding what truly brings joy.

## Genre

It can be considered a **narrative-driven puzzle platformer**. It involves action, adventure, stealth, platforming and combat. The game also focuses on **moral** choices.

## Target Audience

‘Stolen Treasures’ with its vibrant 2D art and a narrative centred around helping others, could be well-suited for a target audience of children. The themes of kindness, adventure, and self-discovery align with content that is both engaging and appropriate for younger players. The inclusion of puzzle-solving elements and platforming challenges can provide a mix of entertainment and cognitive stimulation suitable for this age group. Importantly, there is no violence or explicit bloodshed, making it a safe and enjoyable experience for young players.

## Look and feel

We will be making use of Pixel Art for all of the assets in the game, including the main character, village, forest, dungeon, police, etc. while trying to represent the cultural richness of India. The dialogue in the game will be conveyed to the player using dialogue boxes.

## Gameplay

## **Game Progression**

The game is mainly a story-driven progression where the player progresses by completing the objectives of each level/scene. The progression through levels is linear and all levels have to be completed to complete the game.

## **Level Structure**

Each level is based upon a different aspect of the story and has some associated objective that needs to be completed to progress to the next level. The levels require players to explore the area to complete the objectives

## **Puzzles**

Using our unique mechanics, we will have mechanics-based puzzles which the player needs to solve to progress throughout the game. The puzzles range from interacting with certain items to placing certain things in order and evading traps in various clever ways. These puzzles transform the game from being a simple platformer to something that also challenges you analytically.

## **Enemies and Traps**

Throughout the game, the player will find a variety of enemies which will have different attacking and patrolling patterns. The player will have to skillfully evade them to cross certain areas. Getting caught by an enemy will result in instant death and the player will respawn at some predetermined checkpoints.

The game will also feature a variety of traps such as rotating saws, and spikes. These traps will behave differently in different types of situations. They will have different movement patterns such as they could be going translating horizontally or vertically. They might try to smash you into the ground periodically or even exhibit erratic behaviour. This makes it fun and challenging for the player to navigate through the game and avoid these traps and enemies.

## **Objectives of the game**

Each level will have a different objective that the player needs to complete to progress. The objective will be made clear through interacting with other characters, reading road signs, prophecies etc.

## **Play Flow**

The game progresses from level to level, i.e., once a particular level finishes, immediately the next level starts without any break. The story is conveyed through

interactions with NPCs and cutscenes.

# Mechanics

## Physics

In this 2D platformer, the physics engine is meticulously crafted to encompass fundamental principles such as collision detection, friction, and momentum, contributing to a nuanced and immersive gameplay experience. Collision detection ensures accurate interactions between the player character and the game environment, preventing unintended overlaps and providing a foundation for responsive controls.

Friction is applied judiciously to the character's movements, affecting their acceleration and deceleration on different surfaces. This nuanced application of friction adds realism and strategic depth, influencing the pacing and control precision required in various gameplay scenarios.

Momentum, a key physics concept, is skillfully integrated into the game's mechanics. The player's actions, such as jumping and dashing, influence their momentum, affecting the trajectory and speed of their movement. This dynamic momentum system adds an extra layer of complexity to the platforming challenges, requiring players to master the physics for optimal navigation.

Additionally, the implementation of rigid body dynamics ensures that the character responds realistically to forces applied, enhancing the overall believability of in-game movements. Wall jumping exploits principles of impulse and conservation of momentum, providing players with a satisfying and physics-driven mechanic for scaling vertical surfaces.

Incorporating these physics elements, including collision detection, friction, momentum, and rigid body dynamics, establishes a solid foundation for the 2D platformer's mechanics, contributing to a rich and engaging gameplay experience.

## Movements in the Game

The movement in the game is standard 2D platformer movement. Players can move in all 4 directions using intuitive controls. The movement will have the following aspects

### 1. Responsive Control Scheme:

- Implement a control scheme that provides players with responsive and precise control over the character's movement. Ensure that the controls are intuitive and easy to grasp, enhancing the overall player experience.

## **2. Basic Movement:**

- Introduce fundamental movement mechanics such as walking and running to allow players to explore the game world at a comfortable pace. This serves as the foundation for more advanced movements.

## **3. Jumping Mechanics:**

- Implement a jumping mechanic that takes into account realistic gravity, allowing players to execute controlled jumps with adjustable height. This mechanic is crucial for overcoming obstacles and navigating the environment.

## **4. Dashing Ability:**

- Integrate a dashing ability to add a dynamic element to movement. The dash should provide a burst of speed, enabling players to quickly traverse short distances and evade obstacles. Balance the frequency and limitations of the dash for strategic use.

## **5. Teleportation System:**

- Introduce a teleportation mechanic that allows players to instantaneously move from one point to another. Ensure that the teleportation system integrates seamlessly with level design, offering players both strategic and creative opportunities for navigation.

## **6. Gravity Manipulation:**

- Incorporate a gravity manipulation feature, enabling players to change the strength of gravity within specific areas of the game. This adds a unique twist to the movement mechanics, requiring players to adapt to varying gravitational forces for successful navigation.

## **7. Wall Jumping:**

- Implement a wall-jumping mechanic that enables players to rebound off vertical surfaces. Fine-tune the physics to allow for smooth transitions between walls, encouraging acrobatic manoeuvres and facilitating access to higher platforms.

## **8. Animation Fluidity:**

- Ensure that character animations seamlessly correspond to movement actions. Smooth transitions between different movement states contribute to a visually appealing and immersive gameplay experience.

#### **9. Environmental Interaction:**

- Design the game environment to react to the character's movements. For instance, include elements like moving platforms, conveyor belts, or slippery surfaces that influence the character's movement dynamics, adding variety and challenge.

By addressing these points, the movement mechanics in the game can be diversified, engaging, and aligned with the overall design goals of the 2D platformer.

### **Platformer Controller**

Any good 2d platformer should have fluid and intuitive controls. The player must never have to fight against the controls to navigate through the game. Hence fine tuning each parameter of the physics engine and incorporating conventional techniques used by other good platformers is necessary. We plan to incorporate the following techniques in our player controller.

#### **1. Coyote Time:**

- Extend the character's ability to perform mid-air jumps slightly after leaving a platform. This grace period allows players a moment of forgiveness, promoting a smoother experience by reducing frustration associated with narrowly missed jumps. Coyote Time is crucial for maintaining the flow of the game and enhancing player satisfaction.

#### **2. Jump Buffering:**

- Implement a system that acknowledges jump inputs a few frames before the character lands. This anticipatory input recognition ensures that the jump command is executed immediately upon landing, creating a seamless transition between ground and aerial movements. Jump buffering is especially beneficial in scenarios where precise timing is critical.

#### **3. Input Buffering:**

- Introduce input buffering to the control system, allowing the game to register inputs slightly before they are executed. This minimizes the perception of input lag, making movements feel more immediate and responsive. Input buffering is particularly effective in fast-paced situations where split-second reactions are essential.

#### 4. Transition Animations:

- Create fluid animations that smoothly transition between various movement states. Whether it's shifting from walking to running or transitioning from a run into a jump, these animations visually connect different actions, providing players with a cohesive and immersive experience. Well-crafted transitions enhance the overall polish of the game.

#### 5. Acceleration and Deceleration Curves:

- Fine-tune acceleration and deceleration rates to mimic realistic physics. Gradual acceleration and deceleration provide a more lifelike feel to the character's movements, preventing sudden stops or starts that could disrupt the player's sense of control. These curves contribute to a more satisfying and immersive gaming experience.

#### 6. Air Control:

- Implement responsive air control, enabling players to influence their character's trajectory during mid-air movements. This added control in the air allows skilled players to navigate through complex levels with precision, contributing to a sense of mastery over the character's movements.

#### 7. Consistent Frame Rate:

- Ensure that the game maintains a stable and consistent frame rate throughout gameplay. Fluctuations in frame rate can affect the responsiveness of controls, leading to an inconsistent player experience. A steady frame rate ensures that player inputs are accurately translated into smooth and predictable character movements, contributing to overall control fluidity.

By meticulously implementing these techniques, the 2D platformer can achieve a level of control refinement that enhances player immersion, responsiveness, and satisfaction, ultimately contributing to the success of the game.

## State diagram of the player

### Controls

We plan on using the following controls:

- **Movement** - left using A, right using D
- **Jump** - space

- **E** - Button to interact with interactable objects/characters
- **Esc** - Pause game
- **F** - Teleport
- **X** - dash

## Screen flow

The various screens of the game are

- **Menu Screen** - The game starts with this screen. It gives the player option to start the game and quit the game
- **Game Screen** - This is the screen which shows the game. It will have a small HUD on the top to show objective completion for certain levels such as number of coins collected.
- **Pause Screen** - The player can press the ESC key to pause the game, and then press it again to resume the game, or press the quit button to quit the game.

## Replaying and Saving

There will be no option for saving the game in any menu. The player has to reach certain checkpoints scattered around the level in order to save their progress. The game autosaves when the player reaches a certain checkpoint. If the player dies in a particular level then the player restarts from the beginning of that level.

## Easter egg

We will have a hidden mechanic that we will not reveal to the player which will be the ability to double jump. This will allow the player to press jump another time once airborne allowing for an even greater jump.

## Audio, Music, Sound effects

- **Audio** - There are currently no plans on having voiceovers for the dialogues of characters, just textboxes to depict conversations & thoughts.
- **Music** - We plan on finding royalty-free music for each scene to set the appropriate atmosphere for it if possible. No specific songs or sources have been finalized yet.
- **Sound Effects** - We'll mostly make use of <https://freesound.org/> to get all the sound-effects we need (like gun-shots, throwing items around, grunts, etc)

## **Non-combat and Friendly Characters**

There are only a few non-combat / friendly characters in the game, and they aren't expected to have Artificial Intelligence as they are just parts of the story, and the interaction between the player and the character (if any) is premeditated with no choices given.

# **Story, Setting and Character**

## **Game Narrative**

### **Act 1: The Village**

#### **Introduction**

Introducing Arjun, the thief, living in a vibrant village in the Himalayan region of India. Arjun steals from the rich and distributes to the poor, bringing happiness to a group of underprivileged children.

#### **Conflict**

The families he steals from report to the police and Arjun decides to leave the village to avoid trouble.

### **Act 2: The Forest**

Arjun ventures into a dense forest and discovers a distant temple across a vast valley. Standing in his place, he gazes upon a vast, deep valley stretching out before him, with a small mountain rising in the distance. He notices a sign nearby that boldly declares, "Beyond this lies immense power". Fueled by curiosity and a desire to unveil the temple's mysteries, he leaps towards the mountain, but an unknown force halts his progress. Undaunted, he retreats, contemplating his next move, he takes a courageous leap of faith into the valley.

### **Act 3: The Valley**

#### **Port port fruit**

Venturing through the valley, Arjun unravels puzzles and discovers the Port Port Fruit, giving him teleportation skills. Using this newfound ability, he teleports back to the small mountain, from where he makes another leap into the temple.

### **Act 4: The Temple**



Inside the temple, the air is thick with the essence of the unknown, creating a sense of anticipation and excitement. Corridors lead him deeper and he discovers a dungeon beneath the temple.

### **Dungeon Exploration**

Arjun navigates through the dungeons, solving puzzles and overcoming challenges. Each chamber he conquers leads to another, where he uncovers fruits that grant him special abilities. In the first chamber, he discovers the Aero Aero fruit, giving him control over the wind for swift dashes. The second chamber introduces the Levito Levito fruit, allowing him to become lighter and jump higher. As he approaches the end of the third chamber, Arjun eagerly anticipates the next extraordinary power awaiting him. Finally, hidden at the end, there is a treasure chest that makes him all wealthy.

### **Act 5: The Dilemma**

Arjun emerges from the dungeon, now wealthy and all-powerful. Despite this, he finds himself grappling with a profound sense of loneliness. As he reflects on his life and the pursuits that once brought him joy, particularly the act of stealing, he can't shake the feeling that something essential is missing. A lingering emptiness persists within him. Arjun reflects on his journey, understanding that true happiness comes from helping others and the relationships he formed. He decides to return the stolen treasures to the temple and use his powers for good.

## **Game world**

The game world is set in a Himalayan village. The pixel art style, authentic to classic gaming, brings to life the warmth and charm of Indian traditions, creating a visually immersive and culturally rich game world.

## **Levels**

### **[Platformer 2D][Non-combat]**

- Level 1: The game begins as you're on the verge of entering a house to steal, and the story continues from there.
- Level 2: Escape the village without being caught by the police.
- Level 3: Explore the platform where you land following the leap of faith and locate the Port Port Fruit.

- Levels 4, 5 and 6 **[Dungeon]**: Gather a specified number of airborne coins, discover the treasure at the end to unlock special abilities, and locate the door leading to the next concealed chamber.

# Technical

## Target Hardware

The game being made is neither aimed to be graphically intensive, nor computationally intensive as the most computationally expensive aspect of this game other than rendering the graphics the player controller, which shouldn't be very heavy to simulate. Hence, we feel that the specifications needed for this game would be -

- Operating System - Windows 10 (Previous editions of Windows could pose compatibility issues)
- CPU - Any dual-core processor with speeds of 3GHz or better.
- Graphics Card - A 512MB graphics card.
- RAM - 2GB memory should suffice
- Storage Space - 1GB Storage

## Development Hardware and Software

- Hardware - Just our PCs
- Software - Since this is a 2D game, we will be developing it in Unity.

## Network Requirement

Since this game is story-based, it is an offline game and has no network requirements.