GRIPPY GROOVE

BY

The UnNamed

GAME DESIGN DOCUMENT

VERSION: .01

DATE: 27/03/24

Abishek Ramalingam - 001270035

Anmol Bhardwaj - 001106796

Jack Jordan - 001229919

Table of Contents

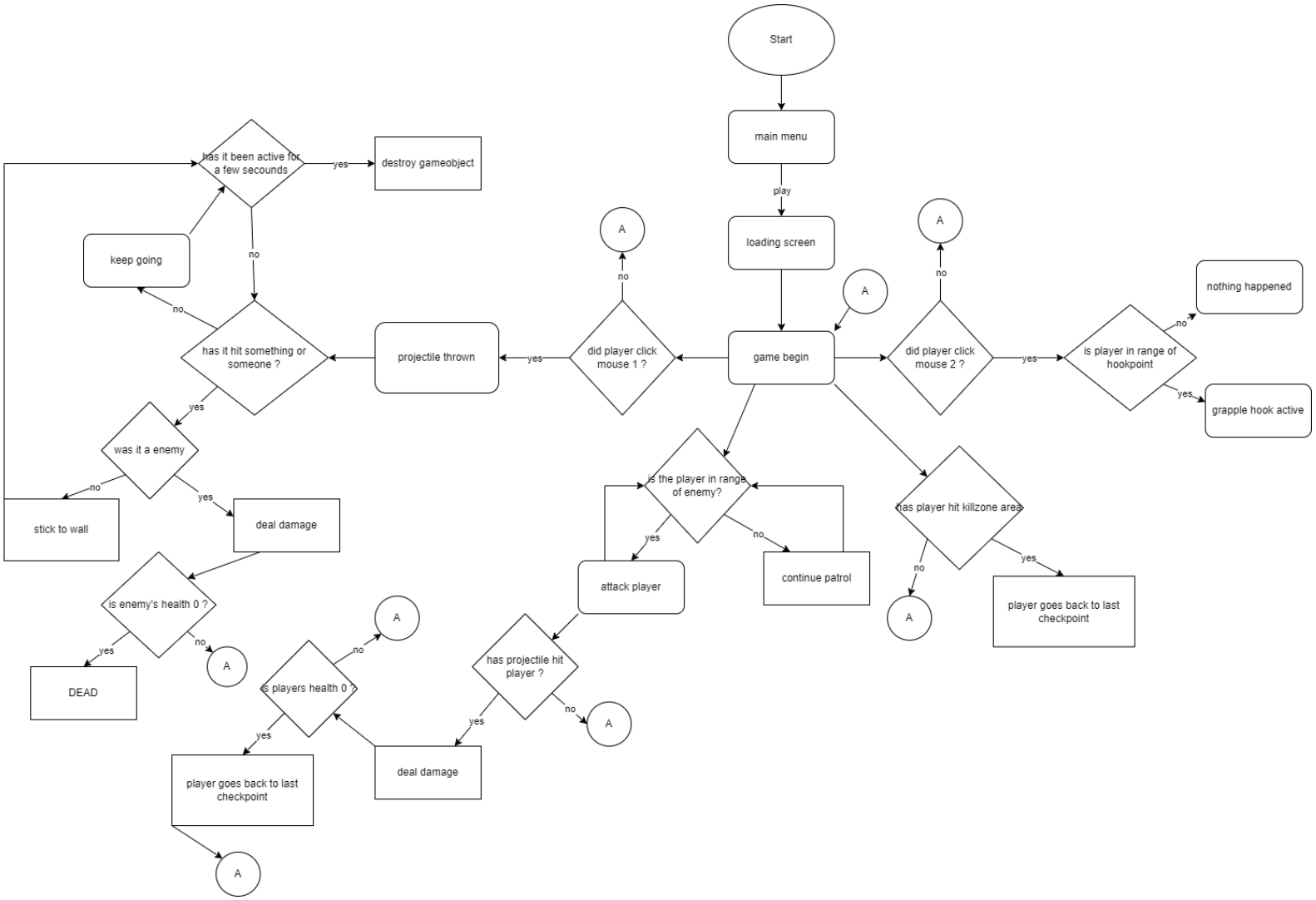
SUMMARY:
With a strong grappling hook at your disposal, go on a treacherous mission as a courageous ninja in "Grippy Groove." Swing across perilous terrain, engage in combat with adversaries, and gather diamonds in order to arrive at the spacecraft, the final objective. Learn to use precise controls and sly strategies to win this action-packed platformer.

Introduction:
Introducing the thrilling and amazing world of "Grippy Groove," where players go on an exciting journey driven by their grappling hook's explosive strength. In this action-packed platformer, players assume the role of fearless ninjas who must traverse a number of dangerous environments in search of fame. Their only weapons are their ninja skills and a reliable grappling hook.
But there are risks at every step they meet. In addition to the dangerous terrain, players must contend with an endless supply of enemies who want to obstruct them at every round. Players must employ every trick in the book to beat and defeat their enemies in order to live, and they must gather diamonds along the way to win the game.

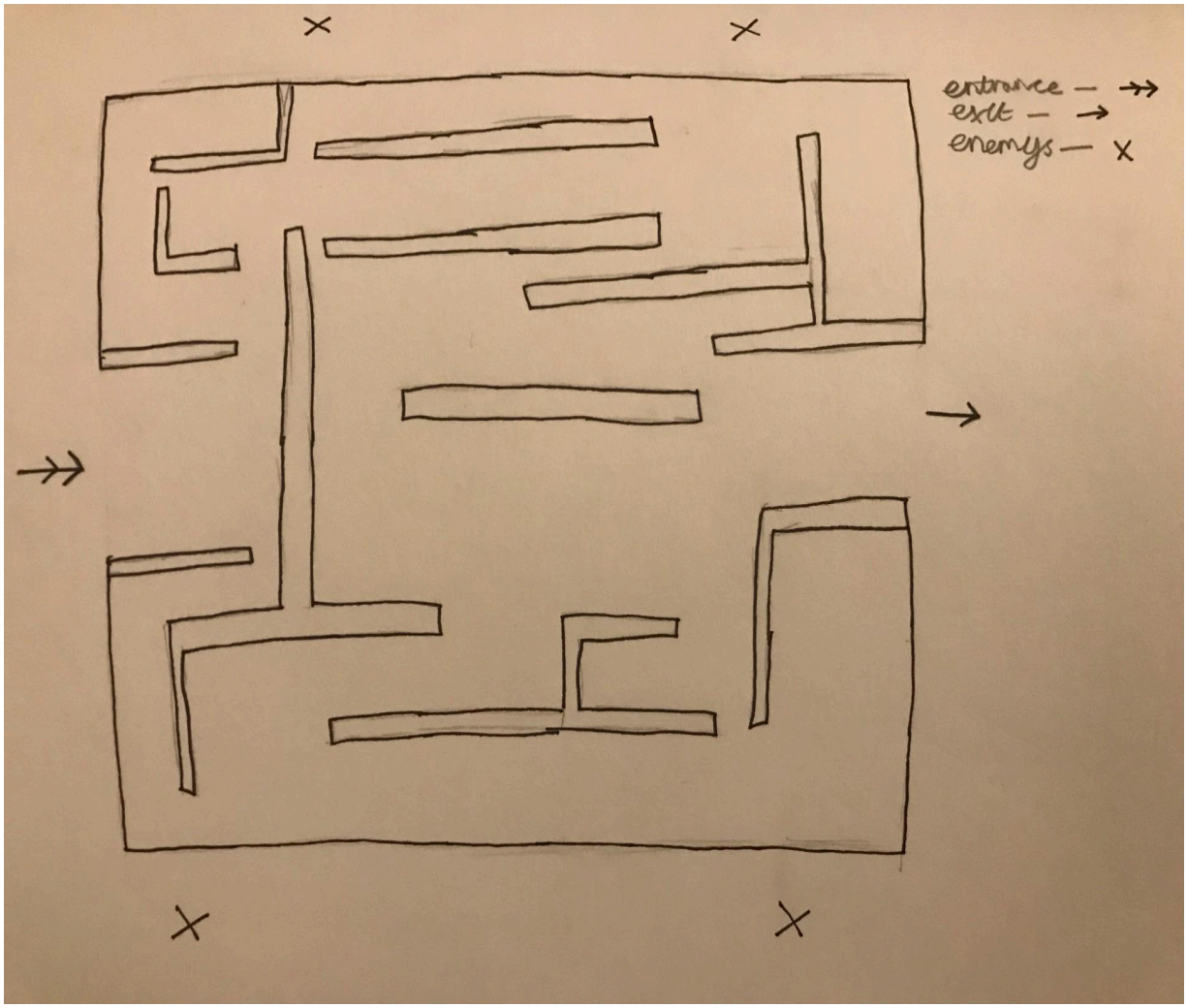
Concept:
In order to beat the game and get back to the land they came from, the player has to explore and make use of all the available areas in the game. This will include the main route of the game which the player will go through by making use of different grappling platforms, lifts/elevators, fighting with enemies who try to eliminate you in various areas of the game, clearing a maze with provided hints, travel through a room full of lasers while also finding ways to reveal hidden grapple points so that they can continue down the main path.
The Grapple or Swinging mechanism is unlike other games and the player is required to use a specific amount of swinging force while they attach their rope to the hook point if they want to get to the next platform without avoiding death by falling into a pool of lava in the volcanic environment. The enemies will be patrolling around different areas of the game and will shoot projectiles at you if you come close enough to their contact range. These vampire-like creatures are smaller in size but can be very deadly if you’re not quick enough to avoid their projectiles or eliminate them before they do the same to you. For some areas of the game, the player will be unable to continue down the main path since there will be no available hook point to grab onto. To encounter this, the player must take a side path and make use of a pressure-based button to reveal a hook point. This button must be held down for the hook point to stay in place and for the player to progress further. The player must

make use of the surrounding objects to keep the hook point revealed so that they can travel across to the laser room.
But before that, they must choose to swing into one of the three rooms in front of them and if they make the wrong choice, they will fall to their death. Upon finding the correct room, they will have to clear the laser room by pressing a switch that will turn the lasers off, allowing the player to proceed further in the game. From here on, they will have to clear a maze with more enemies patrolling around and take a lift higher up for some more swinging and platforming and if they succeed making it to the end with enough diamonds they would have collected on their journey, the player makes it to the time machine and is able to travel back to their own world.

FlowChart:

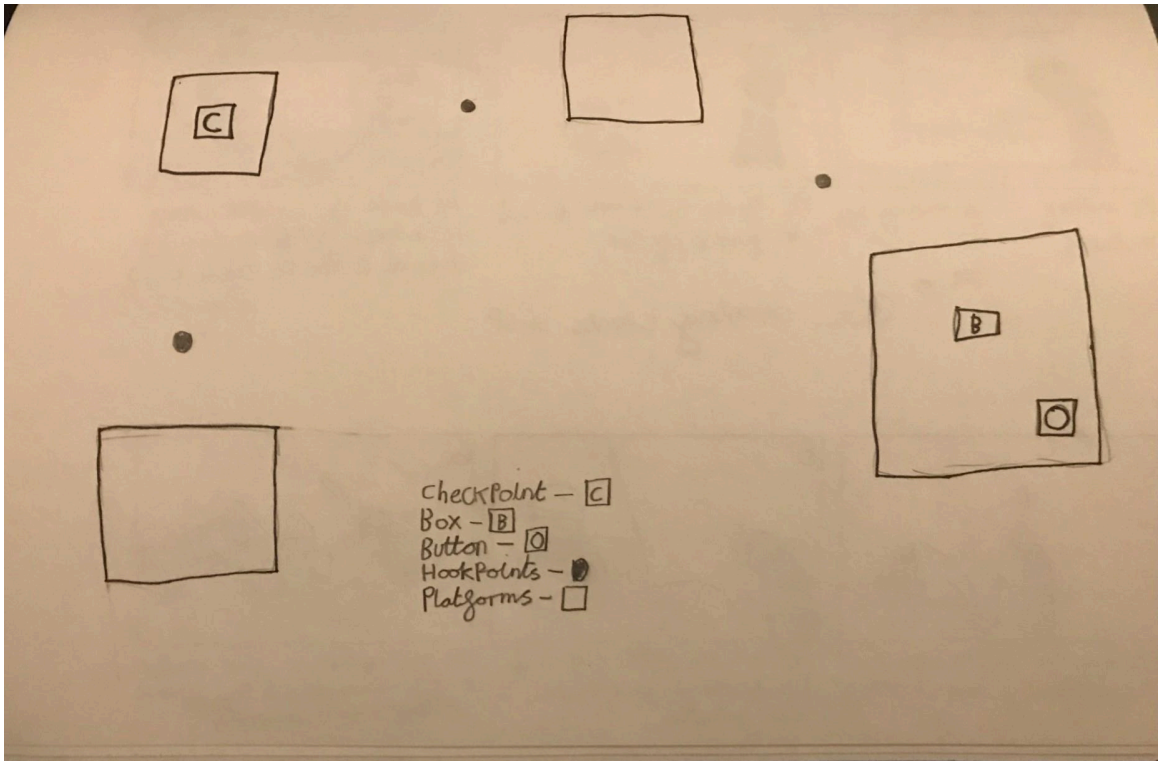


GamePlay:
The player is tasked with collecting all the diamonds scattered throughout the level and reach the spaceship at the end to continue. To do this, the player has to parkour around, using a grappling hook to reach far platformers. Along the way, there will be enemies that will fire on sight, so in order to present a fair playing field, the player can also throw projectiles to fight back. There are different puzzle sections in the level including, a timed sequence and a maze.



The basic platformer with the grappling hook has a more linear way of moving through the level as for one section the player must place a box on a button to keep a hook point active and then should trace back to the previous platform to progress. This is done to keep the

layer from stream lining to the end of the level and finishing too quickly.



The layout of the entire level was done so the player would have to use the grappling hook to progress.
To not make the game too frustrating, there have been multiple checkpoint pads placed throughout the level so the players won't have to rush back to where they were before if they die.
The grappling hook is the main tool that the player will use. It works somewhat similar to the ODM gear from the anime "Attack on Titan" which allows you to hook onto surfaces to traverse more freely, however for our game the movement will be stripped down to only allow forward movement and it can only attach to specific points. The grappling hook is used to get across gaps that you can't just jump across.

Characters:
In "Grippy Groove," players take on the roles of two opposing characters, each with unique goals and attributes. The Ninja is the embodiment of style, flexibility, and cleverness. However, this ninja is special since during the time travel to this dimension a grappling hook got infused into his chest. Equipped with a reliable grappling hook now, the ninja moves through the surroundings with acrobatic grace, dashing through the air and flying from one plane to another. It's time to put his ninja abilities to the test.
These vicious lizards are vampire-like creatures calling their gang the Lires (**LI**zards + vamp**IRE**) and their goal is to protect the valuable diamonds that are scattered over the dangerous areas of the game. The last line of defense against the Daring Ninja's unrelenting pursuit of glory and wealth is these fearsome guardians.

Art And Design:
The game is based in a Volcanic environment with floating platforms that appear to be from a dungeon, it's an odd world and timeline that the player has traveled to. The main area of the game is completely surrounded by these semi-active volcanos and there's plenty of lava at the bottom, it's deadly. The platforms are designed and spaced out in such a way that after figuring out the basic mechanics, the player will have no issues going across the platforms. The player starts at the bottom of the world, close to the lava, and as they progress further, they go higher up in the sky towards the end of the game, away from the lava and closer to their spaceship. This can be seen as levels in the world design as they go higher up each time bringing the total levels to three and each one at an increased height. Volcanic rock texture was used on the mountain plane (game surroundings). As for the textures on most of the walls, various dungeon brick textures were made use of. Finally, for all other objects, materials were created in Unity to fill in additional colors in the game environment such as skybox, lifts/elevators, hook points, etc.

Music and sound:
For the main music in the gameplay, the song 'New Hero In Town' by Kevin MacLeod was used for the chinese-esqt inspired tone, to fit with the ninja aesthetic. It also gives the player an heroic and adventurous feeling when playing the game.
The main menu music is called danger storm by Kevin MacLeod, This sound was chosen to represent the storm that is coming, that being the level and the enemies that wait for the player.
For the end screen music, the song "action-epic-cinematic-victory" by Zakhar Valaha from pixabay is used as it gives the player the feeling they have achieved a great feat, collecting all the diamonds, despite all the dangers.

The sounds used for the player's movements are fast and subtle to match with the character's ninja appearance. The enemies have a hideous laughter when they perish, as if they're trying to mock the player and it makes it clear they have not truly been defeated. The sounds were taken from a wide variety of sources that will be credited in the credit sections.

Controls:
To successfully navigate the dangerous environments and overcome the difficulties that await you in this game, you must become proficient with the controls. Using a mix of keyboard and mouse inputs, players have exact control over the Ninja's moves, swinging, jumping, and grappling their way through each level.
Players may maneuver the Fearless Ninja with agility and accuracy by using the arrow keys for navigation. As for swinging, the main method is the left mouse button, which let players click to throw their grappling hook and glide through the air with unparalleled elegance. In order to release their grasp and advance with the most amount of momentum, players must judge the ideal moment to do so. But control is just as important to swinging as speed. Players may use the right mouse button to launch a lethal projectile assault that will

accurately hit adversaries far away. In order to make sure they are facing the grappling point and to aim their shot precisely, the player must change the camera, which is an important part of this process. Players may learn to manage the swing to maximize their momentum and reduce the chance of overextending and falling to their fate with consistency and precision.
Players need to become proficient at both swinging and projectile assaults, but they also need to learn how to carefully traverse the terrain. They need to make sure that they swing in a forward direction and don't veer off course into potentially fatal obstacles. Grippy Groove players may overcome even the most difficult obstacles and become the ultimate champion by mastering the grappling hook through exact aim, timing, and by being aware of their surroundings.

Progression:
As players move through the difficult stages and conquer many hurdles in "Grippy Groove," the progression mechanism is intended to give them a sense of growth and success. So the players start with the fundamental game controls and mechanics, such as how to swing, leap, grapple, and fight enemies. They pick up the essentials of navigating the landscape and gathering diamonds using the grappling hook.Players have to gather the diamonds that are scattered around each level of the game. These diamonds function as a benchmark for advancement as well as a means of purchasing skills and upgrades
intended to be included in the game at a later time.Gamers' abilities and grasp of the game principles are put to the test as they take on powerful enemy rivals at crucial moments in the game. To win these epic battles, one needs to have quick reflexes, smart planning, and perfect timing. Furthermore, a maze is provided for the players to navigate before they get to the game's finish.
The core aim of the game is to collect the required amount of diamonds and reach the spaceship after defeating the enemies. Once they reach the spaceship the players win and finish their valiant mission as the fearless ninja.

GameMechanics:
The main mechanic of the game is the swing mechanic which will be used to travel across the platforms floating above lava. This mechanic is the main aspect of the game and the world is designed in such a way that it can not be traversed without making use of it. The player also has to figure out that the swinging mechanic requires a mediocre amount of swinging force and not too much or too little since it will result in a setback of continuing from the previous checkpoint again. The movement of the player is fairly fast-paced since he’s a ninja. This mechanic is running using rigidbody in the player movement script which allows for better physics than other ways of controlling the character. For the jump mechanic in the player movement script, vector math was used and swing motion was used for the swing mechanic present. The jump is calculated using rigidbody velocity. The character speed is well balanced but the movement mechanic adds velocity to the player and holds some momentum for how long the inputs were given in the specific direction

which results in the player sliding for a short duration of time relative to how long a directional key input was detected and the player was moved. For having conflicting code for character rotation

The game consists of two kinds of button mechanics to make use of, one being pressure-based to reveal game objects such as hook points and the other one being a timer-based button to turn off lasers for a specific time. The mechanic for revealing the hook point on trigger functions in the inspector where you have linked certain game objects which does this by using a trigger to either set the chosen game object as active or inactive (depending on what the button is set as(door or revelable)). The laser button being timer-based and having been activated by the player, turns off the lasers in a room in the area which allows the player to pass through without taking any damage and progress in the game. However, after a certain amount of time, the timer will reset and the lasers will re-active. This button uses a collision variable to activate and the manager is responsible for handling the lasers turning off along with the timer.

Moving onto the camera, Cinemachine has been used with a rigidbody script instead of using character controller since making use of character controller would result in poor physics mechanics in general and that would go against the purpose of making a swinging game that includes physics. Once set up properly, the cinemachine controller allows a lot of freedom in adjusting the camera to your character in a very simplified manner. The ThirdPersonCam script allows the cinemachine to link with the player as it targets the player's position. Another feature to this mechanic was later added which is camera collision. For this, cinemachine makes it much easier to set that up by using an extension called Cinemachine Collider where you must define the layers you would like to collide again and the ones you would like to ignore. By tagging objects to the desired layers, you can adjust how the camera behaves throughout the map.

Diamonds are the only form of internal economy present in the game and they grant 10 points each and can only be used once. Things like diamond value and volume per object can be managed using the inspector with the help of the diamond script which also keeps track of the total diamond value the player holds.

The Progression in this game requires the use of all main mechanics with Jumping, Swinging, and Running being the crucial ones. As you proceed with the game, there will be a point where you can not grapple onto or swing towards, instead, you have to take a lift to a different level in the game which can be seen as level progression with each time the player uses the lift. The lift is managed using three scripts., The moving script, a parent platform script, and a trigger script. The Platform Moving script can define two or more points and carry the object to and from those points with the help of the parent platform script which detects the collision between the player and the object the script is attached to. Finally, the trigger script calls the Platform Moving script to start moving to defined points after the player collides with the lift.

The Enemies script has many stages to define the behavior of the enemies in the world. This includes things such as patrolling around the area while walking and also chasing the player. The enemy will enter the chase state when in Sight Range which can be adjusted in the inspector. Along with this, things such as the health of the enemy, attack range, and choosing who to attack can also be adjusted through the inspector. This script also makes use of the health bar script to indicate a visual health point bar that indicates the health of the enemy after being hit by player projectiles. The same script is also used by the player to show their health status. However, With the amount of public variables, you can use the inspector to freely adjust a lot of aspects of enemy behavior resulting in being able to use the same script for different kinds of enemies with different statistics.

The projectiles use two different prefabs, one for the player and one for the enemies. Both work the same as the other with the only difference being who gets damaged by it. The projectile spawns in front of the player/enemy and flies forward until it hits a wall, where it will stick to and disappear after a while, or when the projectile hits a target described in the scripts. The player and the enemy can throw a projectile as per the described fire rate and projectile speed in the inspector by using the right mouse click button (defined as “KeyCode.Mouse1” in the script as throw Key). The OnHit script manages the projectile's collision and damage.

In terms of maneuvering, due to how the player movement script functions, there are things like air strafing in the game which helps around the core mechanic of the game i.e swinging. Character rotation was conflicted and was not properly resolved so the animations used here are adjusted according to how the player maneuvers around in the game. With the player's speed being fairly adjusted to the speed of incoming projectiles from the enemies, the player can easily dodge them if they are as quick with their reflexes as a ninja. Air strafing plays a big part in making it safely to the next platform if the player is swinging toward it as the jump velocity carries the player in the script.

Other scripts that are responsible for player death are KillZone, and Laser. Killzone script straight up kills the player as they trigger it when they come in contact with the box collider on the object attached to the script. The laser does a specific amount of damage to the player and eliminates the player eventually when they take enough hits. Both of these environmental mechanics will respawn the player to their recent spawn point.

To save progress during the game, there are several checkpoints present in the game which, after triggering using a collider, will be the player’s new spawning point if they get eliminated. The checkpoints also have an indicator feature for when you’ve successfully activated the checkpoint by changing its color. This script is managed by the Checkpoint Manager script which keeps track of all the present checkpoints in the game and which one the player has recently triggered.

For managing all the player animation scripts, animationStateController script is used to define all the keys and conditions for when specific boolean animations will be triggered. Conditions such as inAir, isGrounded, and detection of the ground collision were also added as additional code for smoother and more accurate animation triggers. Some more adjustments to the code were made for the player to trigger the falling animation when falling down the platforms without having contact with objects tagged as ground.

UI:

HUD(Heads Up Display**)**

Diamond count:

The amount of diamonds the player has collected is shown clearly by the diamond count function. This enables players to monitor their advancement and determine their proximity to accomplishing the game's goals. Players gather diamonds, and they count in real-time.

Health Bar:

The player's health bar, which shows how much health the player character has left, is clearly visible on the HUD. Throughout gaming, players may keep an eye on their own health thanks to this visual depiction. The health meter accurately indicates the player's current condition as it drops when they take damage. Gamers are able to make well-informed choices and modify their tactics accordingly. Players may decide to play more conservatively and prioritize evasion and recuperation over aggressive maneuvers if their health is poor. On the other hand, a player with a high health level could choose to play more aggressively, taking calculated risks to get past opponents and barriers.

Main Menu:

Play Button: Initiates gameplay and takes the player to the loading screen.
Settings Button: Opens the settings menu where players can adjust various options.
Quit Button: Exit the game.

Loading Screen:

Provides visual feedback that the game is loading.
Displays controls of the game in pictures, helping players familiarize themselves with the gameplay mechanics before starting.

Settings Menu:

Resolution Dropdown: Allows players to select their preferred screen resolution for optimal visuals.
Graphics Dropdown: Enables players to adjust graphical settings such as quality and effects.
Volume Slider: Controls the overall volume level of the game's audio.
Fullscreen Toggle: Allows players to switch between fullscreen and windowed mode for their preferred display setup.
Back Button: Returns players to the main menu

Pause Menu:

Resume Button: Allows players to continue the game from where they left off.
Main Menu Button: Returns players to the main menu without exiting the game.
Quit Button: Exits the game entirely.

Settings Button: Provides access to the settings menu, enabling players to adjust options even while in-game.
By integrating these components into our menus, we ensure that users may interact with "Grippy Groove" in a smooth and intuitive manner, which enhances the immersive nature of the gaming experience. A well-designed interface makes it easy for gamers to customize their gaming, which raises player satisfaction and engagement levels. Also, the use of visual aids on the loading screen effectively prepares players, enabling a rapid understanding of the controls and notions. This calculated move guarantees player happiness and strengthens the game's accessibility, making it simple for both experienced players and novices to completely lose themselves in the gripping adventure of "Grippy Groove".

Technical Requirements:

Operating System: Windows 7/8/10 (64-bit), macOS 10.12+, or Linux equivalent.
Processor (CPU): Intel Core i5 or AMD equivalent.
Graphics (GPU): DirectX 11 compatible graphics card or equivalent with at least 2GB of VRAM.
Memory (RAM): 8GB or higher.
Storage: At least 2GB of free space for installation.
Input Devices: Keyboard and mouse, or compatible gamepad/controller.
Display: Minimum resolution of 1280x720 pixels.
Unity Version: Developed using Unity 2022.3.7f1 LTS (Long Term Support) or later for stability and compatibility.

Additional Requirements:
Internet connection for initial installation and updates.
DirectX 11 or OpenGL 4.5 support.
Audio output for sound effects and music.

Testing:

During the testing of the game, the most crucial part was figuring out where the hook points must be positioned in order for the playtesters to make it to the required platforms. During the initial development stage, the player movement and swing scripts worked together in such a way that it allowed the player to increasingly add swing velocity by using directional key inputs which resulted in a swing more that was more than required to transverse the platforms and the player would fly off the map or collide through other objects. Changes were made multiple times to the movement script which resulted in a different travel range for the swing mechanic and so did the hook point since they had to be adjusted accordingly. After playtesting with these better-adjusted grapple points, some other platform size adjustments had to be made for better platforming by the players. Along with the hook points, the rooms were re-arranged in a different order than it was planned in the initial stages as having most of the rooms and platforms to the way they are now had better game progression and difficulty scaling according to each stage in the game.

A few playtests also revealed that some players were able to access further hook points than they should, which was the result of the hook range being high. To counter this, barriers were also added to the game environment in order to prevent the players from getting to locations they should not be able to access without going through the main pathway of the game. Initially, the lift mechanic was made to travel only from point a to b, but as players were failing to stay on this moving platform, changes were made to make it function as an automatic lift once triggered, which now would travel from a to b and back to a. Along with these changes, the size of the platform was adjusted along with box colliders on it to have smoother collision detection and trigger the scripts.

For the animations, as the character rotation was not working properly due to conflicts with the player rotation code in the player movement script and ThirdPersonCam scripts, different animations for each directional input were assigned in the animation controller script. The animation transitions were correctly adjusted to match the speed and maneuvering of the player. With the player not having the trigger to transition properly from certain animations to falling animations, changes were made to the script by adding more “if statements” and variables to manage the logic of the animation transition better. After testing these changes for a while, the falling animation was now working whenever the player was falling off the platform and when dropping down after jumping.

Other changes such as making the rooms in “guess the correct room” (area right before the laser room) were made by adjusting their size and tagging them to the proper layers and tags resolved minor issues being displayed with animations flickering before these changes were made. The maze room was initially designed for enemies to shoot down at you from heightened platforms while the player tries to figure out the way through the puzzle. This however could not have worked due to the projectile shooting angle relative to the design of the maze and enemy positioning. So the enemies ended up being added inside the maze, maneuvering around looking for the player instead. With maneuvering, the player had an issue of bouncing on the platforms when letting go of the swing with a lot of velocity which was fixed by adding roughness material onto the player.

Some kill zones were adjusted and some were added through all stages of the game, the initial kill zones were added on the roofs of rooms in the lower areas where the player can fall onto if they fail the platforming above. However, this area is inaccessible and so the kill zones were added to those objects so that the player is sent back to the checkpoints when colliding with areas that result in elimination. The world environment structure did not have any colliders up until the final stages of the game and was using a lava-colored kill zone with the trigger box collider.

Along with this, changes to the camera positioning were also made as during the initial stages of the game, camera collision was not added and cinemachine collider was used to add this feature. The issue here was the camera focusing at the pivot point of the player (NijnjaFBX skin in Unity) which was located at the feet of the player. To overcome this issue,

an empty game object was added as the player's sub-category and was adjusted to its back area, and then set this new empty game object as "Look at" in the cinemachine collider settings.

WHO DID WHAT:

Abishek	Anmol	Jack
Scripts: • Player Movement • Health Bar • Enemy • Control Audio • Diamond • MainMenu • PauseMenu • Swing • Manager • Audio Manager • Settings Menu • FollowCam 3D Modelling: • Hook tip • Initial Level idea UI: • Main Menu • PauseMenu • Credits • Settings Menu • Scoreboard Audio: • Death • Lift • Lazer • Hooked	Scripts: • AnimationState controller • Movement • ParentPlatform • PlatformMoving • ThirdPersonCam • TriggerPlatform 3D Modelling: • Whole Level • Spikes & Blades • Boxes • Volcano • Spaceship • Level Re-arranging and Design • Textures Audio: • Enemy	Scripts: • Buttons • CheckPoint • CheckPointManager • EndscreenButtons • EnemyOnHit • Killzone • Lazer • Lazer button • Projectile • Spaceship Win • onHit • LoadingBar UI: Loading screen Audio: • Start • Scene • Enemy Death • Footsteps

CopyRights:

Brackeys (2020). How to make a HEALTH BAR in Unity! YouTube. Available at: https://www.youtube.com/watch?v=BLfNP4Sc_iA. [Accessed 25 Feb. 2024].

Dave / GameDevelopment. (n.d.). FULL 3D ENEMY AI in 6 MINUTES! || Unity Tutorial. [online]
Available at: https://www.youtube.com/watch?v=UjkSFoLxesw [Accessed 24 Feb. 2024].
Brackeys. (2017, December 6). SETTINGS MENU in Unity [Video]. YouTube. Retrieved from https://youtu.be/YOaYQrN1oYQ.

www.youtube.com. (n.d.). *Unity Rigidbody Player movement 3rd and 1st Person in 2 minutes*. [online] Available at: https://www.youtube.com/watch?v=bqLRWTkgNGg [Accessed 3 Apr. 2024].

www.youtube.com. (n.d.). *THIRD PERSON MOVEMENT in 11 MINUTES - Unity Tutorial*. [online] Available at: https://www.youtube.com/watch?v=UCwwn2q4Vys&list=LL&index=27&t=290s [Accessed 3 Apr. 2024].

www.mixamo.com. (n.d.). *Mixamo*. [online] Available at: https://www.mixamo.com/#/?page=1&query=ninja&type=Character.

www.mixamo.com. (n.d.). *Mixamo*. [online] Available at: https://www.mixamo.com/#/?page=1&query=vampire&type=Character

www.youtube.com. (n.d.). *ADVANCED SWINGING in 9 MINUTES - Unity Tutorial*. [online] Available at: https://www.youtube.com/watch?v=HPjuTK91MA8&t=302s&pp=ygUQZGF2ZSBzd2luZyB1bml0eQ%3D%3D [Accessed 3 Apr. 2024].

www.youtube.com. (n.d.). *Make an Elevator in Unity Tutorial*. [online] Available at: https://www.youtube.com/watch?v=cW-5JYZLlvQ&list=LL&index=7&t=2s [Accessed 3 Apr. 2024].

Credits:

Animations and models - Maximo
Ninja model - https://www.mixamo.com/#/?page=1&query=ninja type=Character
Enemy model - https://www.mixamo.com/#/?page=1&query=vampire&type=Character
Animations - https://www.mixamo.com/#/?page=2&query=&type=Motion%2CMotionPack
Health bar and UI reference - Brackeys - https://www.youtube.com/@Brackeys
Swinging script and enemy AI reference - Dave - https://www.youtube.com/@davegamedevelopment
Third Person Follow Cam - Dave - https://www.youtube.com/watch?v=UCwwn2q4Vys&list=LL&index=27&t=290s
Elevator - The Game Dev Core
https://www.youtube.com/watch?v=cW-5JYZLlvQ&list=LL&index=7&t=2s
Player movement reference - TDS
Audio - Darren Lloyd, pixabay- https://pixabay.com
Evil laughter - https://pixabay.com/sound-effects/search/evil%20laughter/
Player movements - Creator - Leohpaz, Made on - May 21,2022, downloaded from OpenGameArt.Org
Special thanks to andrei(for all the fixes),Peter and Joaquin
Textures -Volcano - https://wallpaperaccess.com/lava-texture
Floor - https://www.wallpapertip.com/wdown/mwRmo_grey-brick-wall-background/
Maze Walls - https://artpictures.club/autumn-2023.html
Maze Floor - https://www.pinterest.ca/pin/639792690804474411/
Lazer Room - https://ches.artstation.com/projects/lVzwRa