

The background is a stylized, blocky illustration of a night scene. On the left, there's a stone building with a green wall. In the center, a large, dark stone head is visible. The ground is covered in green grass with some yellow patches. Several glowing, rectangular platforms are scattered around, and a torch is in the foreground. The overall style is reminiscent of a video game or a digital art piece.

Wildman Adventures

Design Document

Daniel Brooks

Contents

High Concept	4
One line Introduction	4
Theme/Environment/Setting	4
Unique Selling Point and Key Mechanics	4
Gameplay Summary.....	4
World Summary	4
Character Summary	4
MoSCoW	5
Must Have.....	5
Should Have.....	5
Could Have	5
Won't Have	5
Gameplay Loop	6
Gameplay Loop	6
Game Pillars	7
Player Controls, Characteristics and Abilities.....	8
Controls and Map.....	8
Character Abilities	9
Unreal Engine 5 Movement Stats	9
Characteristics and Stats	9
Camera	12
Mechanics, Hazards, and Puzzles	13
Game Mechanics.....	13
Mechanic 1.....	13
Mechanics 2	14
Mechanic 3.....	14
Mechanic 4.....	15
Mechanic 5.....	15
Game Hazards	15
Hazard 1	15
Hazard 2	16
Interactions and Pick ups	17
Coins.....	17
Chests.....	17



Life.....	17
Game Initialization	17
Mechanics Flow and flowcharts.....	18
UI.....	20
Font	20
Player Wireframe	20
Menu Wireframes	21
Game Conditions.....	21
How do you win	21
How do you lose.....	22
Art	22
Asset List	22
Meshes	22
Character.....	22
Environment.....	22
SFX.....	22
Music.....	22
UI.....	22
Level Design Documentation	23
Goal of Level and Key Areas.....	23
Story.....	24
Puzzle Interaction.....	24
Level Influences.....	25
Level Design Plans.....	27
Level Flow and Beats.....	28
Level Location.....	29
Area 1	29
Area 2	30
Boulder Run	31
2D Puzzle Section	32
Critical Paths.....	33

High Concept

One line Introduction

Wildman Adventures is a 3D puzzle platformer where the character is in control of the 'The Wildman' with a goal of collecting coins and completing the level.

Theme/Environment/Setting

WA (*Wildman Adventures*), will feature linear levels that make the player go from A to B. The overall theme of the level will be bright colours and a stylized theme. Inspiration for the game comes from Mario 3D World, Crash bandicoot and Spyro.

Unique Selling Point and Key Mechanics

WA will have classic platforming mechanics and interesting puzzles. The game will utilise Nintendo's design philosophy of introducing a mechanic slowly building on that mechanic and eventually leading to player mastery.

Key mechanic, Platforming Elements – WA will be looking to appeal to the old school platforming genre fans by leaning heavily into the puzzles and mechanics of the level having unique mechanics for the player to learn and have fun with. The puzzle mechanics of the game will be the main driving force of player progression through the levels.

Gameplay Summary

The player will start at point 'A' (the start of the level), as the level will be linear the overall goal of the game will be to make it to point 'B'. The player will also have to collect coins a 'side' element to the game in order to better complete the level. Whilst collecting coins isn't necessary for completion, failure to do so will result in a worse score at the end.

Throughout the level, players will have to learn different puzzle mechanics to progress through. The puzzles will be introduced, combined, twisted and finally mastered; this leads a difficulty being increased the further the player progresses through the level.

World Summary

The Wildman finds himself on these floating island and platforms with one goal, to collect coins and leave the world.

Character Summary

The Wildman will be the main character the player controls. He is a cartoonish tribesman with the goal of collecting coins and making his way through the levels. He will have the ability to walk, jump, double jump, and sprint.

MoSCoW

Must Have

- Complete Game Loop
- Mechanics Puzzles
 - At least 1 well designed mechanic that is present through the map and twist/ challenges player ability.
- Traps
 - At least 1 trap that works well as a twist, with a mechanic.
- Rewards
 - Coins
 - Chests
 - Health Pickups
- Playable Character (Must Be)
 - Third Person Perspective
 - Limited camera rotation/movement
 - Life Loop (Life, death respawn)
 - Modular
 - Different de-coupled parts that can be taken off or added to other character to still work with game.
 - Usable by other designers to edit or change character.
 - Boulder Chase
 - Unique level aesthetic and lighting.
 - Controller support/built for

Should Have

- Game sound and Music. Sperate/ different ones for menu and gameplay.
- Game juice and feel affects:
 - Controller haptic feedback
 - Camera shake
 - Particle effects
 - SFX
- Another puzzles/ mechanic that is present throughout the level.
- Another trap that works well as a twist to the mechanics.
- Cinematic Intros and ends.

Could Have

- Multiple Levels
- Level Streaming elements
- More reward variety
- Ai attack elements
- Player attack abilities
- In game player cutscenes.
- Colour blind assistance.

Won't Have

- Coop play
- Multiplayer Online
- VR support
- Custom dialogue sounds

Gameplay Loop

Gameplay Loop

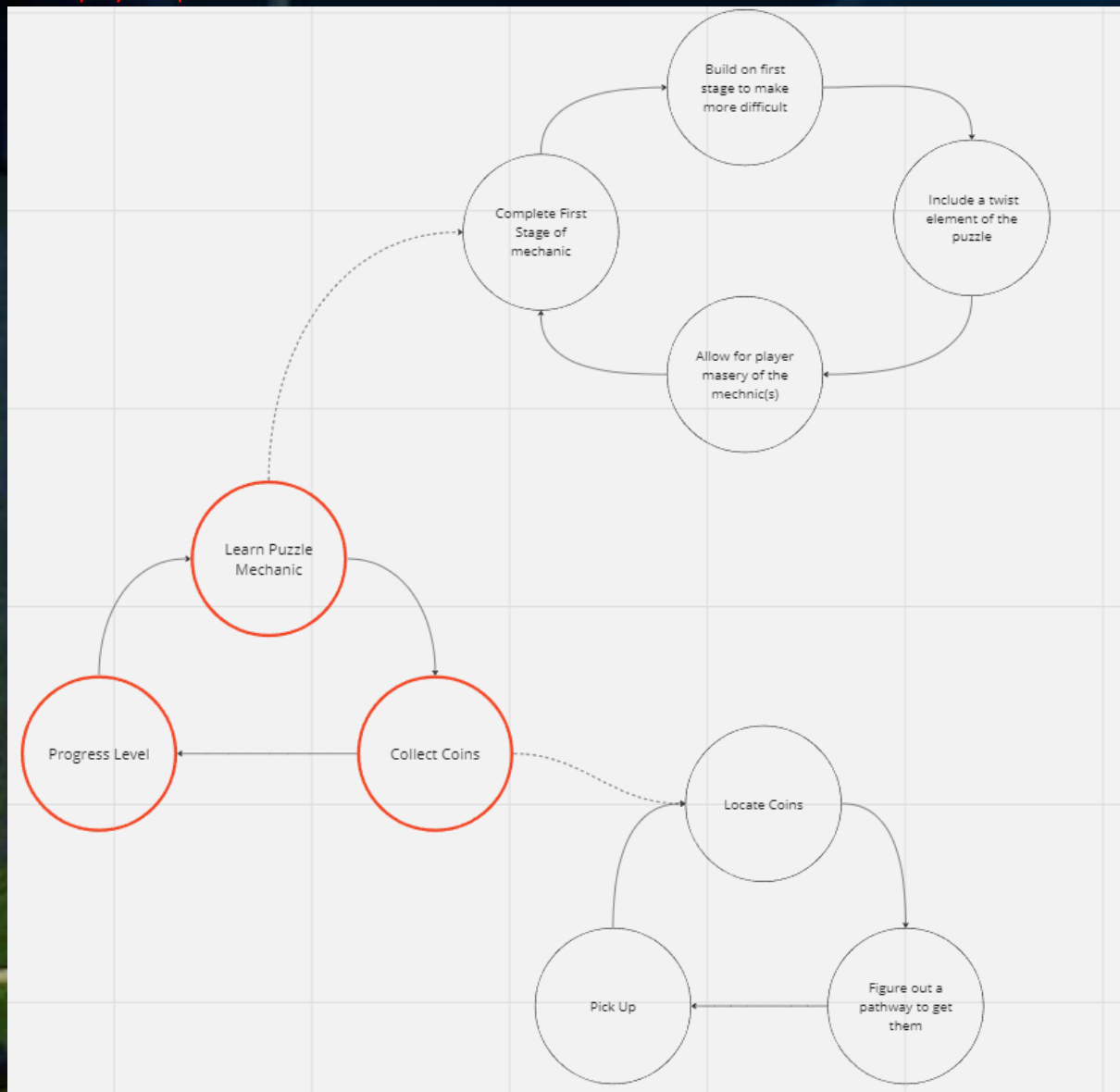


Figure 1 - Main Gameplay Loop (Red), with side loops

Game Pillars

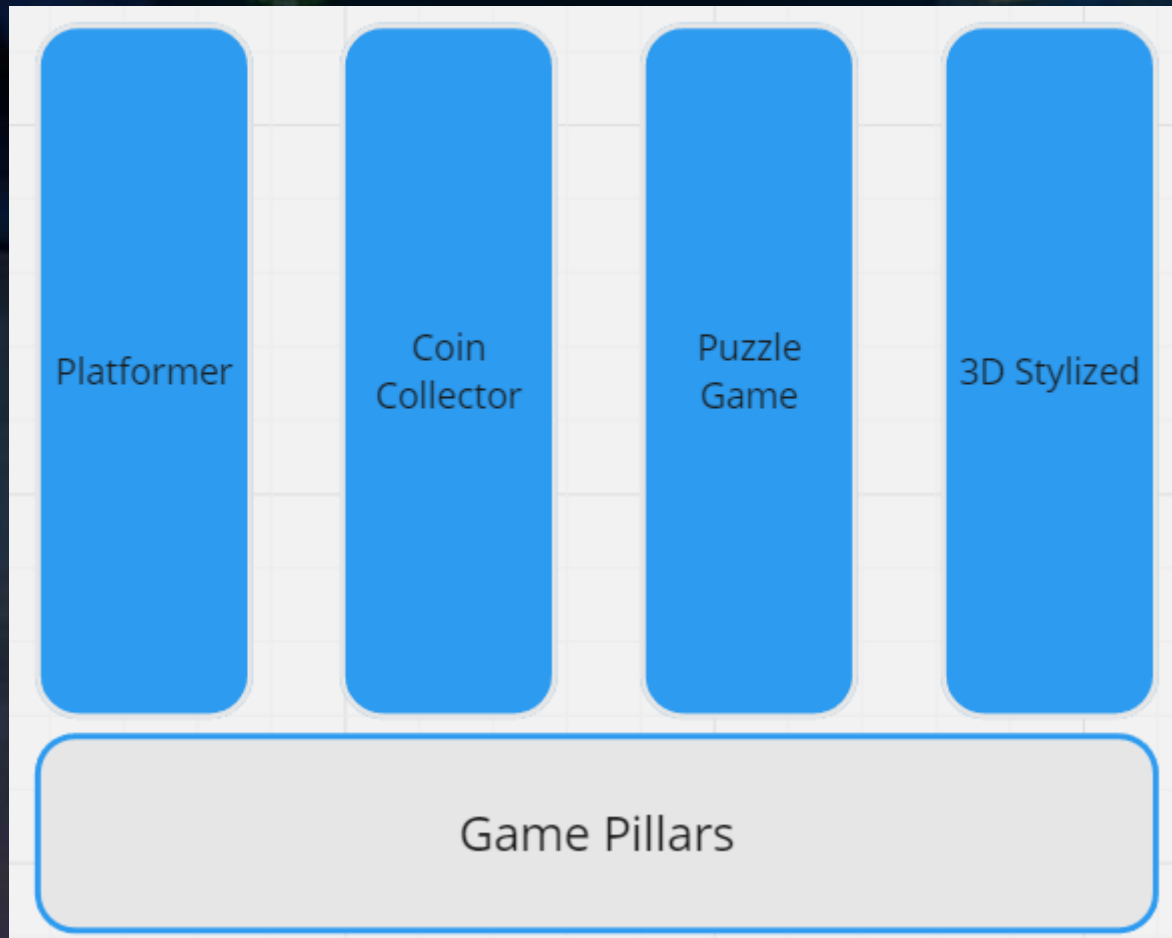


Figure 2- Game Pillars

Player Controls, Characteristics and Abilities

Controls and Map

Controls for the character on PC and Controller (

Xbox

Button	Interaction
Left Analogue Stick	Movement/Navigation and direction
Bumpers (RB, LB)	Rotate Camera
X	Interact
A	Jump. Press Twice to double Jump
Menu	Pause and bring up menu

PC

Button	Interaction
W, A, S, D	Movement/Navigation and direction
Q, R	Rotate Camera
E	Interact
Space Bar	Jump. Press Twice to double Jump
ESC, P	Pause and bring up menu



Figure 3 - Control scheme for controller

Character Abilities

- Movement (Walk/Running)
- Sprint
- Jump
- Double Jump
- Interact
- Death
- Camera Rotation

Unreal Engine 5 Movement Stats

- Acceleration - 750
- Walk Speed - 600
- Walk Speed (Sprinting) - 800
- Gravity Scale – 2.75
- Jump Velocity - 900
- Air Control – 0.9

Characteristics and Stats

The player will have many different components added to the character that allows it to function correctly. This also means the characters is very modular in design and can be broken, attached and created to do other things. Components needed: Com_(Component_)

- Com_Camera – controls the player camera and camera events that change the player camera throughout the level.
- Com_Coins – An inventory that stores coins collected.
- Com_HapticFeedback – Holds all the events for firing off haptic feedback events for controllers.
- Com_Health – Keeps track of the player health as well as healing done and damage to the player.
- Com_Interaction – Allows for interaction of the player to objects in the world. All events are done in this component.
- Com_JumpBlock – Controls the need to fire off events for jump blocks in the level.
- Com_Respawn – Controls the respawning of the player and re-setting of objects in the world.
- Com_Sound – Fires off any sound that needs to play that isn't played in world space or needs spatialization.
- Com_UI – Updates the current UI on screen with updates for coins collected, health and chests found.
- COM_Endgame – Used to set character events and animations specific to the character.

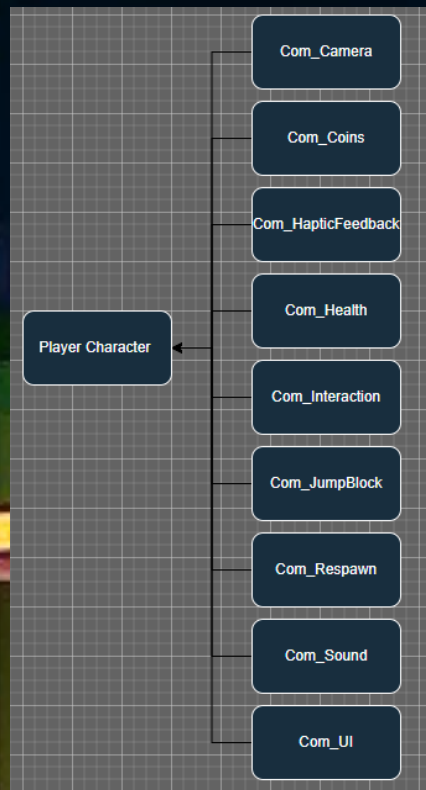
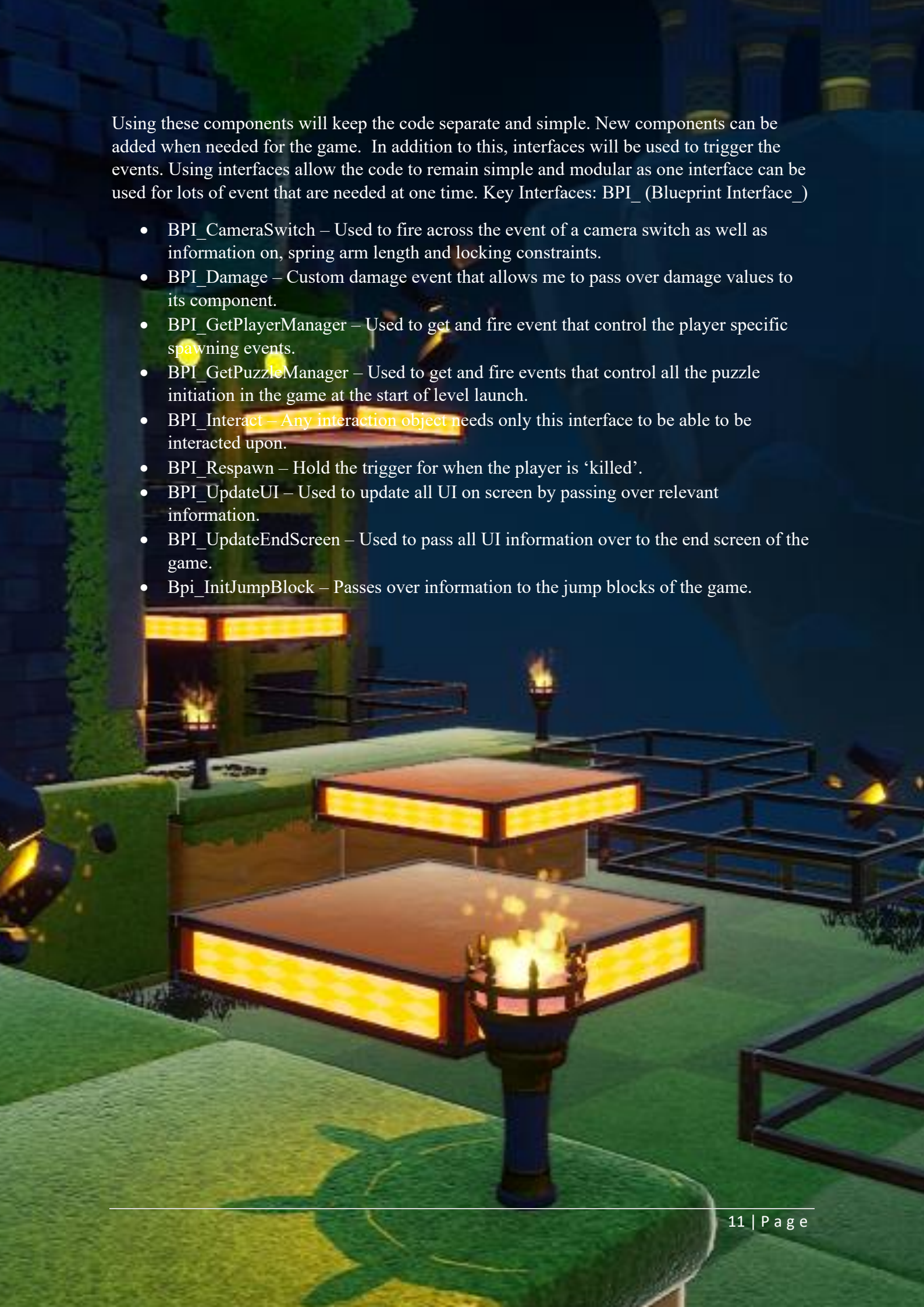


Figure 4 - Character Component List



Figure 5 - Player Jump Flowchart



Using these components will keep the code separate and simple. New components can be added when needed for the game. In addition to this, interfaces will be used to trigger the events. Using interfaces allow the code to remain simple and modular as one interface can be used for lots of event that are needed at one time. Key Interfaces: BPI_ (Blueprint Interface_)

- BPI_CameraSwitch – Used to fire across the event of a camera switch as well as information on, spring arm length and locking constraints.
- BPI_Damage – Custom damage event that allows me to pass over damage values to its component.
- BPI_GetPlayerManager – Used to get and fire event that control the player specific spawning events.
- BPI_GetPuzzleManager – Used to get and fire events that control all the puzzle initiation in the game at the start of level launch.
- BPI_Interact – Any interaction object needs only this interface to be able to be interacted upon.
- BPI_Respawn – Hold the trigger for when the player is ‘killed’.
- BPI_UpdateUI – Used to update all UI on screen by passing over relevant information.
- BPI_UpdateEndScreen – Used to pass all UI information over to the end screen of the game.
- Bpi_InitJumpBlock – Passes over information to the jump blocks of the game.

Camera

The player camera will be placed behind the player initially and locked in place. The player won't be able to 'freely' the camera around and instead, the camera can be rotated around the player at a 90-degree angle by pressing the appropriate button. After this, the camera is locked in position again until rotated. The player forward direction is driven by the camera angle, meaning the way the camera is facing is forward.

There will specific 2d sections of the game that will add more flair to the game. During these parts the camera will be lowered to a vertical height of the player and locked from rotating. This is ended by the player completing the part of the puzzle. This will feature like Mario games that use the gimmick of 2D parts in a 3D engine.

During the boulder run, the camera will face the player character and be locked from revolving. The camera will be increased in height and have a special mode linked to the component part that will set the camera to a certain height and arm length.

The camera will need to show off the world and puzzles ahead of the player so they player can learn what is ahead of them and master it. This is paired with the camera have a direct 'look' at the player as/ the player character will be the focal point of the player screen rather than world or puzzles.

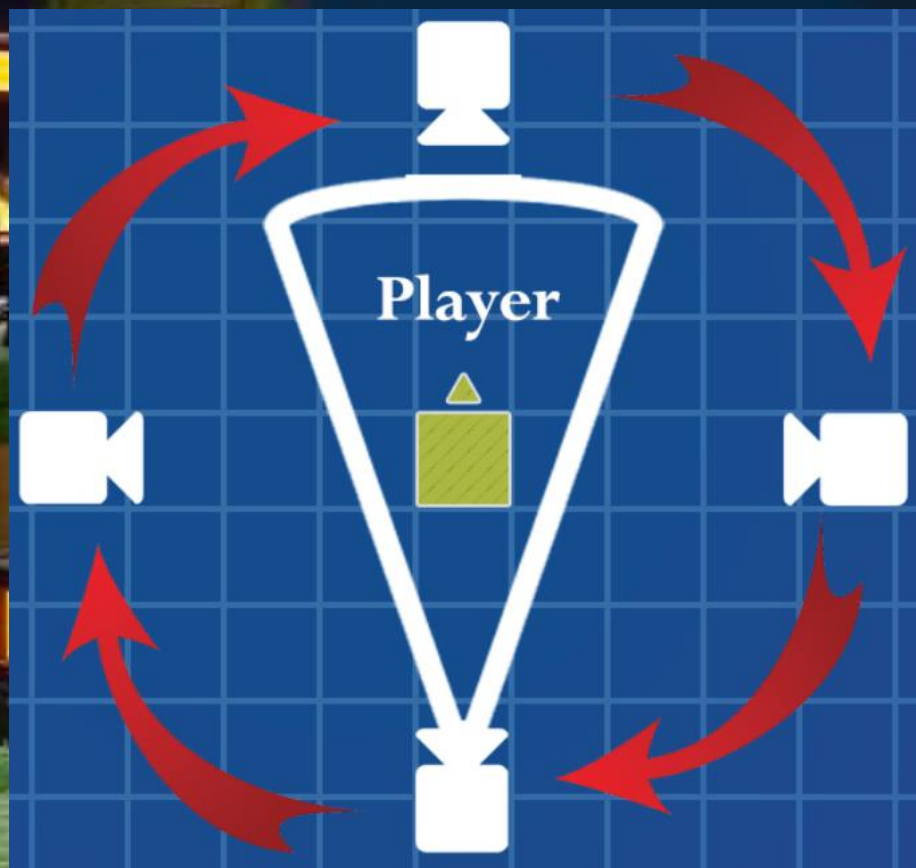


Figure 6 - Camera Angles and locations

Mechanics, Hazards, and Puzzles

Game Mechanics

Mechanic 1

Jump Pad

Features

- The Jump Pad is a block that has two states available that can be selected during development with each state being a unique to how it works.
- First State:
 - The first state makes it function that works off the players input for jumping. When the player jumps, the block alternates between being visible (solid collision) and invisible (no Collision).
 - This means the player can only jump on the block that is currently visible to them.
 - The block can be set as visible or invisible at spawn, and it'll switch first to whatever on it isn't at that time.
- Second State:
 - The second state makes the block function on a countdown timer that alternates between being visible and invisible.
 - The countdown lasts several seconds and at the end will switch block to either visible or invisible.
 - This means the player can only jump on the block that is currently visible to them.
 - The block can be set as visible or invisible at spawn, and it'll switch first to whatever on it isn't at that time.

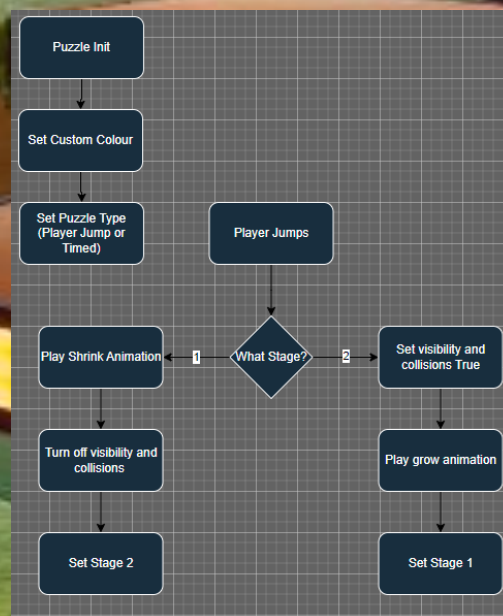


Figure 7 - Jump Pad Flowchart (First State)

Mechanics 2

Falling Platforms

Features

- The falling platform is a block that only allows the player to stand on it for several seconds before becoming invisible and unable to stand on until the block resets.
 - When the player stands on the block, it starts to change colour indicating change is about to happen.
 - When the block is a second away from becoming invisible, it begins to shake indicating that the block is near 'destruction'.
 - If the block disappears, it will reset after several seconds and become usable again.
- The falling platforms will make the player have to move quick and make fast decisions, to progress and not fall off.

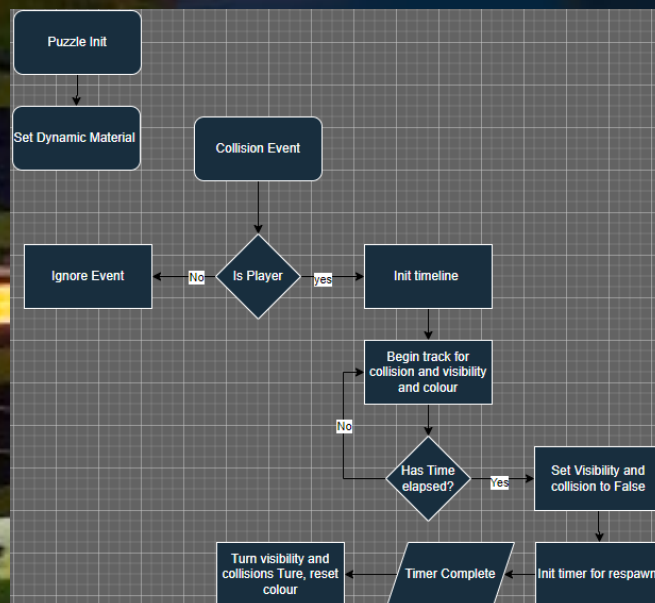


Figure 8 - Falling Platform Flowchart

Mechanic 3

Revolving Platform

Features

- The revolving platform is a block that revolves on any axis from its centre point.
- This means the block can be used to access an area or block it.
- This mechanics has two states that it can be in as well, (State A and B)
 - State A get the block revolving from the game start.
 - It revolves once on a 180-degree pattern. Once it has reached its 180-degree goal, it will stop revolving and pauses for 3 seconds.
 - In this time State B is activated and does the same thing.
 - This means both states alternate between on another.

Mechanic 4

Moving Platform

- The moving platform is a mechanic that allows the player to move from point A to point B.
- The platform has two switches that make the platform travel in its given direction (forward and back backwards).
- The platform is mainly used as a mechanic too allow the player to travel from one point to another instead of the player having to run or use movement mechanics.

Mechanic 5

Boulder Chase

- The boulder chase will be the final element of the level. This will act in very similar structure to crash bandicoot's boulder chase.
- Once the trigger has been triggered by the player a boulder will move towards the player and it is the player's job to run away from the border whilst jumping across ledges and past puzzle mechanics to make it to a safe area.
- If the boulder touches the player, it causes instant death and they're respawned back at the closest checkpoint.
- The boulder chase will culminate the level and use past mechanics as difficulty also increases.

Game Hazards

Hazards are used as mini game mechanics that are designed to make one of the main mechanics more difficult to complete (adding a twist). Hazards can damage the player, causing the player to lose life. All damage dealt by hazards are equal to 1.

Hazard 1

Shockwave

- The shockwave emits an expanding ring that if the player touches, deal one damage to the player.
- The shockwave has a warmup time before it emits the ring. This will be clearly visible to the player so they may brace themselves or be ready for it.
- This is designed so the only way for the player to not get hit by the wave is to jump to avoid it. This coupled with other jumping mechanics of the game will give the player something to think about went traversing the level.
- Different speeds can be entered to either increase or decrease the expansion time to make the hazard harder or easier.

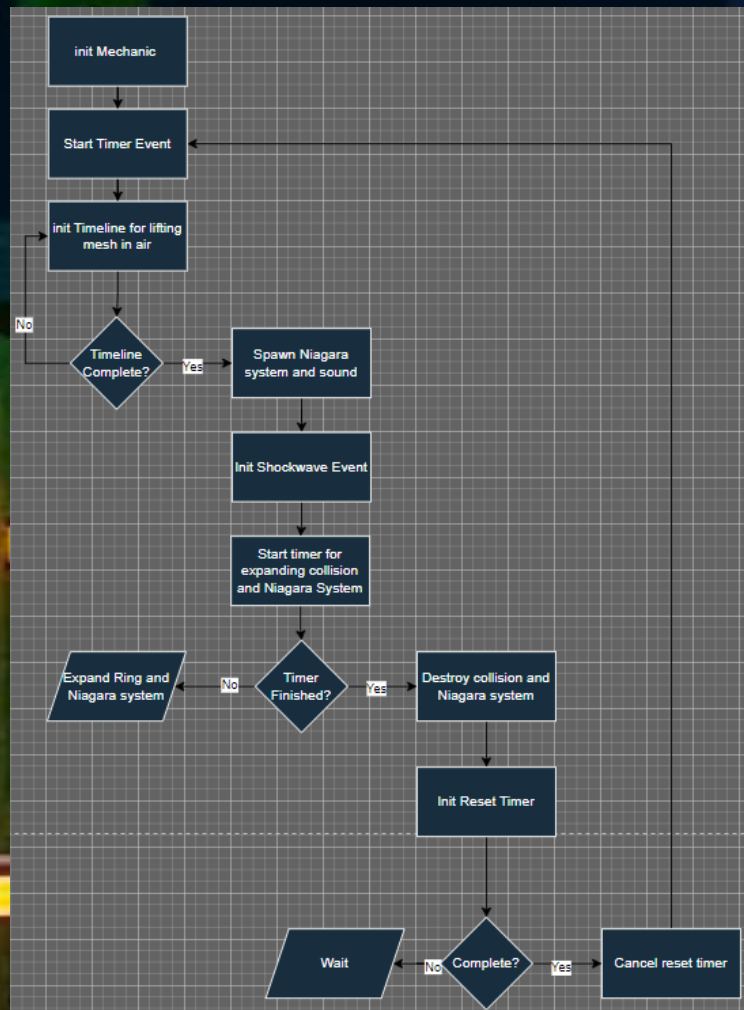


Figure 9 - Shockwave Hazard Flowchart

Hazard 2

Rolling Beam

- the rolling beam is a beam that moves from point A to point B and one straight line.
- During the movement, if the player contacts the beam, it deals one damage to the player.
- This hazard has two states that can be selected, one way by the beam is destroyed at the end of its path and is respawned at the start, and another whereby it ping pong's between point A and point B.
- Different speeds can be entered to make the beam move faster or slower depending on how difficult the part needs to be.
- The idea behind this hazard is it will make the player have to make a quick decision or else risk getting hit by the beam.

Interactions and Pick ups

Coins

- The main thing the player will be looking for whilst progressing through the level will be coins. Coins will act as a guidance to guide the player for the level as well as an optional pickup that players will want to go find more of.
- Once a coin has been picked up by the player via an overlap, the coin will be added to the players inventory and permanently remain there. This also means that coins are not lost on death and are therefore, not needed to be respawned upon death.
- The total number of coins collected will be shown at the end of the level, this will give the player an incentive to collect all of them in order to 100% the level.

Chests

- Chests are more rare objects that are placed in the world and require the player to do some sort of challenge to access them.
- This means that the chest is more of a reward for the player once they complete a hard action.
- Chests contain coins, that when opened, send the coins flying in the air and an animation. These can then be picked up by the player like a normal coin.

Life

- Heart capsules can be found throughout the level as well, please heal the player for one point of damage.
- Whilst they are not part of the main progression of the game, they allow for the player to keep moving or are those a helpful utility for those who aren't as good at the game.
- Players will not be penalised for picking up these hearts.

Game Initialization

To keep the game as modular as possible and clean, I will be looking to create a system that spawn and initializes everything in a modular 'chain' that can be chopped and change depending on the player and designers' choice, e.g. adding a starting level sequence or spawning and starting all the main puzzles of the game. To achieve this all the necessary events, need to fire on after each over after completion all leading to the player being spawns and possessed. Initialization chain needs:

- Game Manager Init
 - Spawn level cutscene – plays the intro level cutscene.
 - Spawn player character – spawns the player character in game (not possess though)
 - Spawn level puzzles – Spawns and initialize all puzzles of the game.
 - Spawn game UI – spawns in the main player UI
 - Spawn game sound and music – starts playing the game music and sounds.

- Play game – once chain has finished, possess the character and grants control of the character.
- *Endgame sequence* – triggered at endgame. Spawns' endgame UI and music.

Mechanics Flow and flowcharts

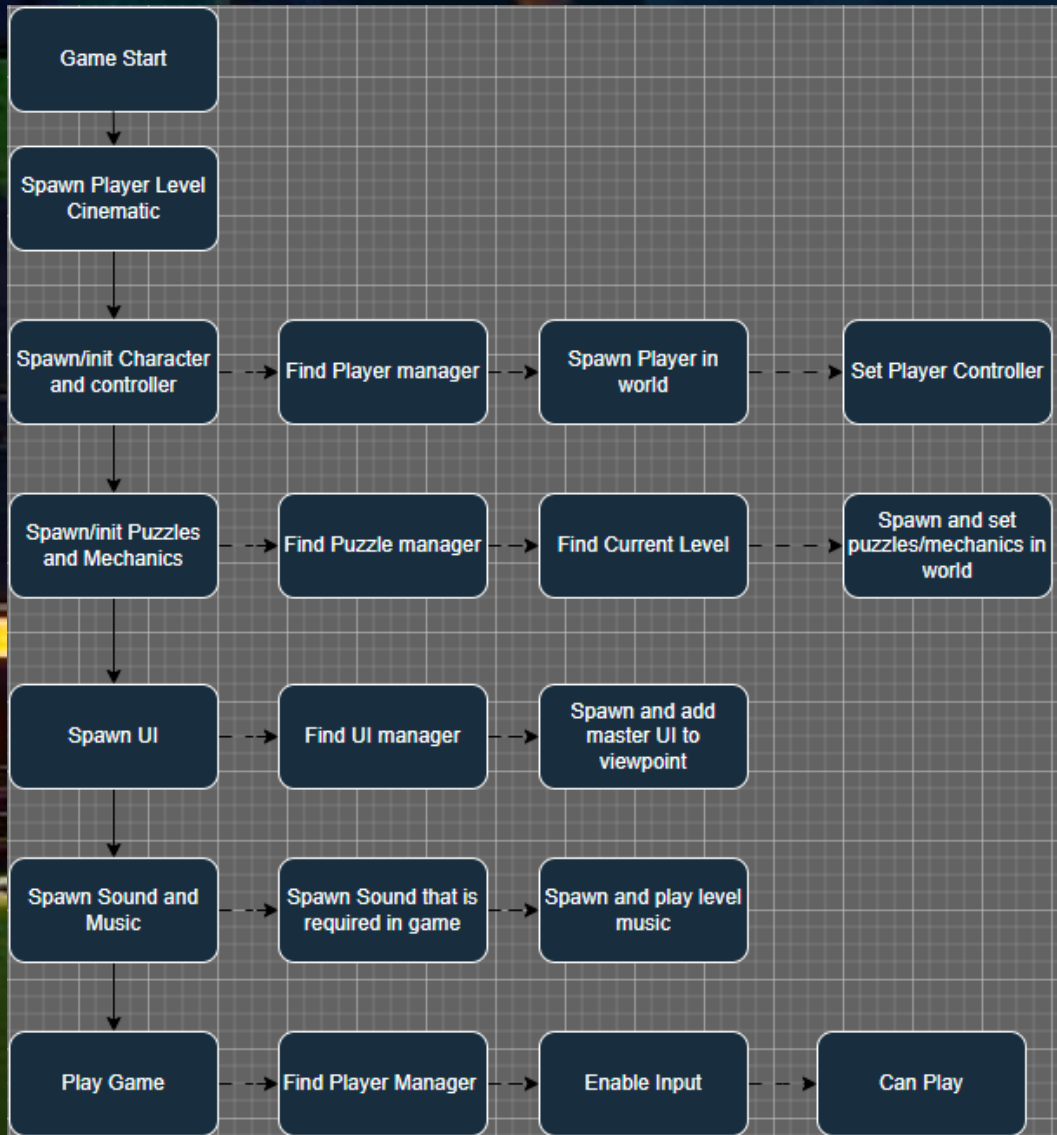


Figure 10 - Level start initialization chain

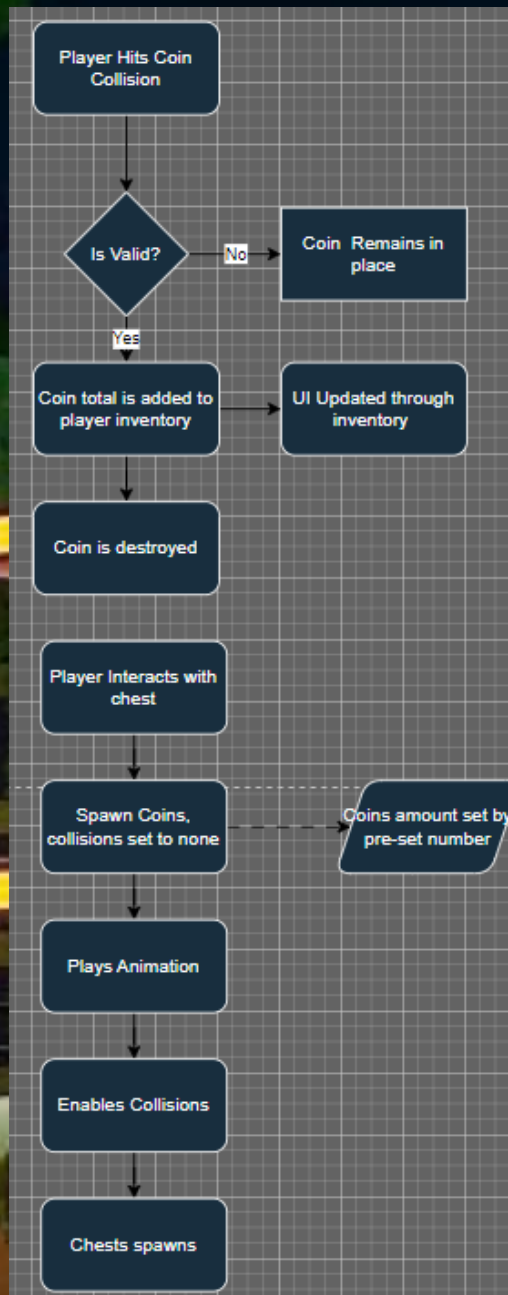


Figure 11 - Coin Flowchart

UI

Font

- <https://fontmeme.com/fonts/maybe-coffee-font/>
- <https://fontmeme.com/fonts/want-coffee-font/>

Player Wireframe

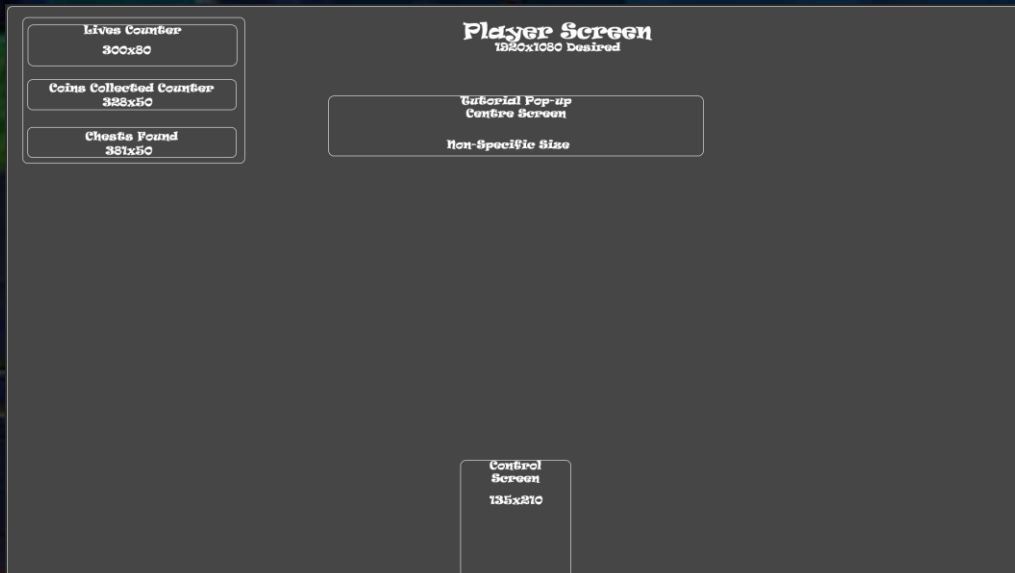


Figure 12 - Player Wireframe

The player's UI will be rather minimal during gameplay, this is to avoid over cluttering the screen distracting the player from currently gameplay. The main elements that will be present on screen for the player to see will be:

- A control map - this shows what the buttons are for gameplay. The current controls that will be present are:
 - Rotate Button
 - Interaction Button
 - Jump Button
- UI text popups - these will be present in the middle of the screen and gives the player helpful advice and tips on what to do at a certain point in the level. These should be quick, fly onto the screen and leave.
- Level counters - there will be a level counter at the top left of the screen, this will keep count of:
 - Coins collected and maximum number to find.
 - chests opened and maximum to find.
 - Current number of lives.

Menu Wireframes

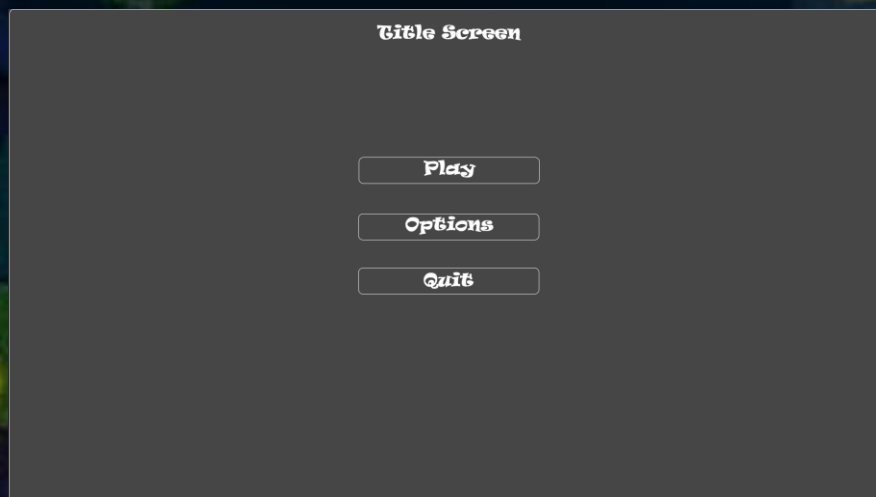


Figure 13 - Menu Wireframe

For menu and menus of the game will have three main options for the player to choose these being:

- Play/Resume – starts and/or resume gameplay for the player.
- Options – Opens up the options for the game for the player to adjust. Options include:
 - Sound – Sliders that can adjust the music volume, sound effects and overall volume of the game.
 - Controls Scheme – An image of the control scheme of the game for player to take a glance at.
- Quit – leaves the game if on the Main menu or quits to the main menu if on gameplay.

Game Conditions

How do you win

The player can 'win' by making it to the end of the level. Once the player has reached the end of the level scores of the level are added up these include:

- Coins – The number of coins found throughout the level.
- Chests – The amount of chest found throughout the level.
- Overall Progression – a total sum that shows how much of the level did the player complete (finding coins and chests). Calculation being –
 - Total Number of Coins Found + Total Number of chests Found.
 - Divide total number above by Max Number of coins and chest that can be found.
 - X100 rounded up.
 - E.g. 50 coins found + 3 chests found = 53
 - 53/ 68 total coins to be 'can' found = 0.779
 - 0.779 x 100 = 78% (Rounded up)

How do you lose

- The player loses by losing all lives and dying. There isn't any permanent death, instead the player will respawn at a checkpoint and are able to continue again.
- No coins or chest collections are lost upon death.
- The only way the player can lose is if they don't make it to the end of the game.

Art

Asset List

Assets for the game will be sourced from either the unreal marketplace or asset sellers online. All relevant links will be added in this section of the design document. Whilst the asset pack may be listed this doesn't mean that I will be using all assets available from that pack.

Meshes

Character:

- <https://www.unrealengine.com/marketplace/en-US/product/wildman>

Environment:

- <https://www.unrealengine.com/marketplace/en-US/product/unreal-engine-hour-of-code>
- <https://www.unrealengine.com/marketplace/en-US/product/dreamscape-nature-meadows>
- <https://www.unrealengine.com/marketplace/en-US/product/platformer-starter-pack>
- <https://opengameart.org/content/teleporter-circle>
- <https://www.unrealengine.com/marketplace/en-US/product/stylized-sky-dungeon>

SFX

- <https://www.unrealengine.com/marketplace/en-US/product/puzzle-games-vol-1>
- <https://www.unrealengine.com/marketplace/en-US/product/interface-item-sounds-pack>

Music

- Jason Brooks - 'Main game music theme'
- Rolando Gomez - <https://www.fiverr.com/musicalroly>

UI

- <https://www.unrealengine.com/marketplace/en-US/product/cartoon-clean-gui-ui-kit-56-widgets-90-icons-712-png>

Level Design Documentation

Goal of Level and Key Areas

The level(s) of the game are designed to be short, fun, puzzle pieces that build upon the Nintendo design philosophy of introducing a mechanic and building on it until the player mastery of that mechanic. The rules for building levels in the game need to follow the following rules:

Mechanics:

- Introduce a mechanic in a 'safe' environment.
- Allow the player to develop their skills of that mechanics.
- Add a twist to the mechanic, challenging the player a challenge when solving the puzzle.
- Allow the player to show mastery of the mechanic, in a 'finale'.
- Clear rule set to the mechanic

Level:

- Short in length of map, high in energy/pace.
- Each 'area' or 'level' must feel and seem unique to show how different a puzzle or section is.
- Must have some 'side tracks or hidden areas of the map for the explorer player. This can then lead to the reward of 'chests.
- Needs to be completable with the player leaving with a sense of excitement about finishing the level. *play testing required.*

The levels need to feature POI and lights to guide the player around the map. Whilst the puzzle will be able to lead the player around the map to a certain extent, this could become boring or repetitive. Having rest areas around the map where the player can take a break from platforming for a bit will be required to break up gameplay. This can also be seen as a chance to introduce mechanics in the safe areas. Rewards such as coins or chest can be granted for players who choose to explore around these parts, however they can't deviate away from the players main critical path to much or else the player could, lose where they are at, what a mechanic does or simply might not fit the style of a game that I am making for the project.

For this reason, the player needs to be able to stick to the critical path as much as they can. Puzzles and mechanics should keep these parts interesting enough that the okayer wont lose interest in the game.

Story

The dumb, nomadic man known as only 'Wildman' is in search of ultimate power and knowledge. The Wildman must complete a series of levels and puzzles to acquire this forbidden knowledge he so desperately wants.

Puzzle Interaction

The table below shows a puzzle matrix that will help me in choosing what puzzles interact at what moments and gauging if the result was interesting or not. This can be used to come up with many different levels for the game, making each one unique and special compared to each other. This can be useful as, so I don't get lost or confused with what mechanics does what and how it interacts with others. More puzzles can be added to the list and tried to see how of an interesting result occurs when they interact with each other.

Puzzle	Jump Pad	Falling Platform	Revolving Platform	Moving Platform	Shockwave	Rolling Beam
Jump Pad	----- ---	Makes the player panic and have the move quickly again. Fun and useable.	Can cause the player issues where they jump when they didn't need to or mean to. Potential.	Player can wait and jump to switch pad. Needs to be paired with hazard.	Makes the player must think about their next move	Makes the player choose between dodging or finding a new platform
Falling Platform		----- ----- ---		Interesting mechanic with the player having to time the drops to landing on platform. Potential.	Gives the player two different pressures time and dodging	Puts the player under pressure via the use of two-timed mechanics
Revolving Platform			----- ----- ---	Platform doesn't always sink up and without being in safe zone. Can	Shockwave can pass player. Not much pressure is applied at this section.	Player is given comfortable time to didge the beam

				become boring.		
Moving Platform				----- ----- --	Player must jump which make platform stop. Can become very stop start.	Makes the player must think ahead about their next move. Viable with more development time.
Shockwave				----- -		Both the player dodges. Not much puzzling or platforming happens.
Rolling Beam						----- -----

Level Influences



Collage of reference images I will be using to light and make my game look and feel like. Using a mix and match of colour schemes, level layout and turoialisation. Games include.

- Mario 3D world
- Crash Bandicoot

- Luigi Mansion
- Spyro



Level Design Plans



Level Flow and Beats

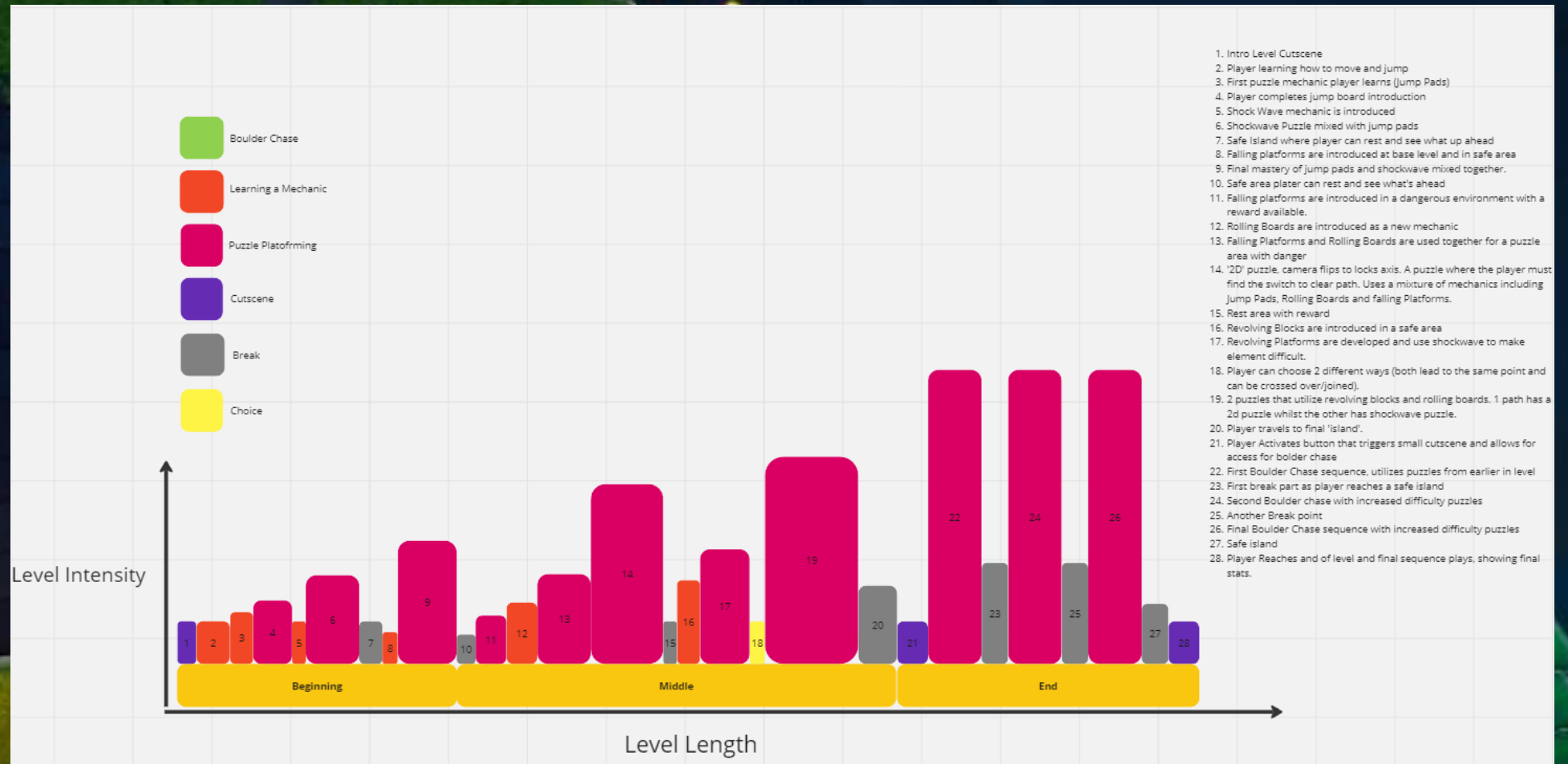
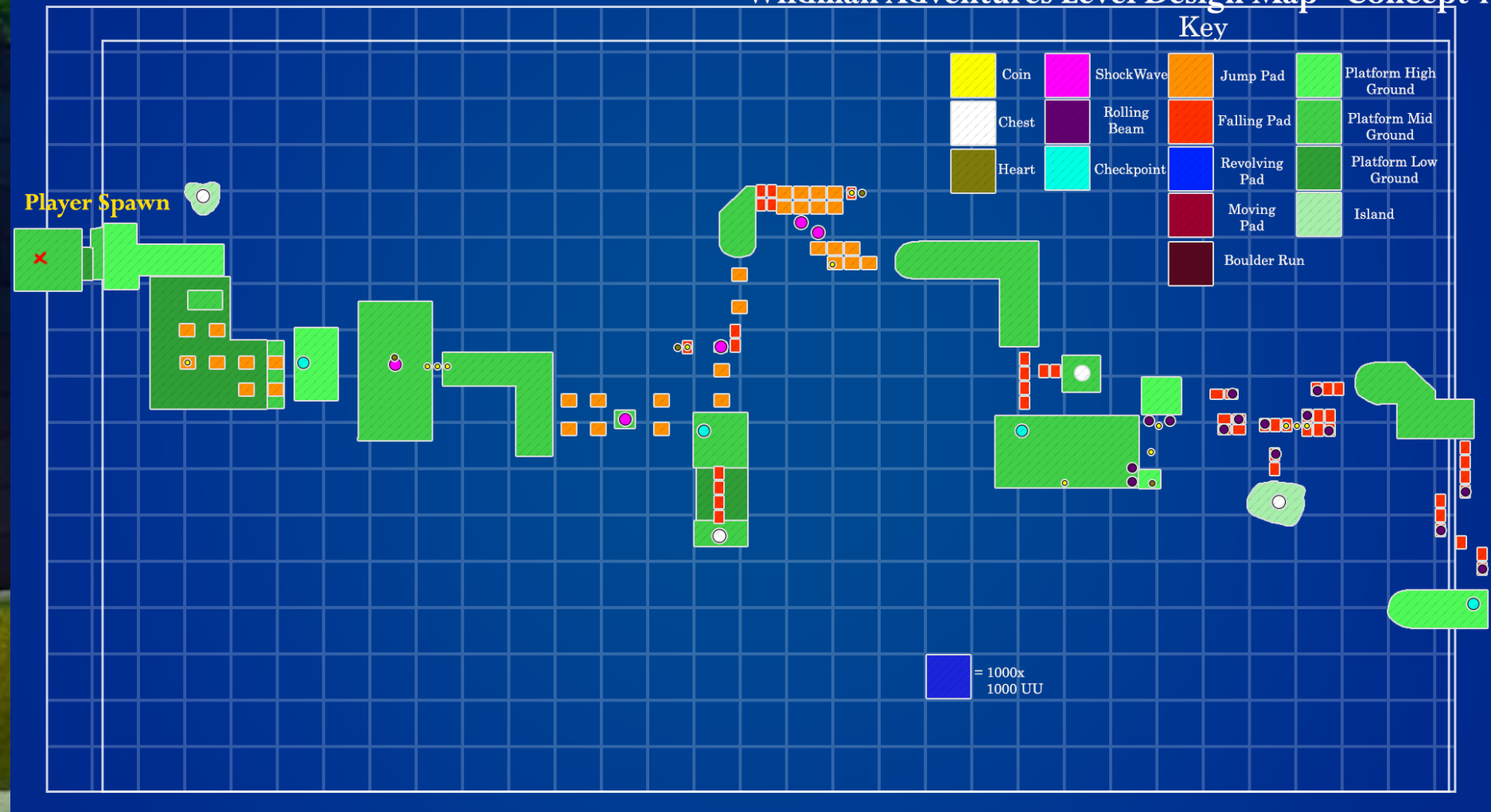


Figure 14 - Level flow and pace

Level Location

Area 1

Wildman Adventures Level Design Map - Concept 1



Area 2

Wildman Adventures Level Design Map - Concept 1

Key

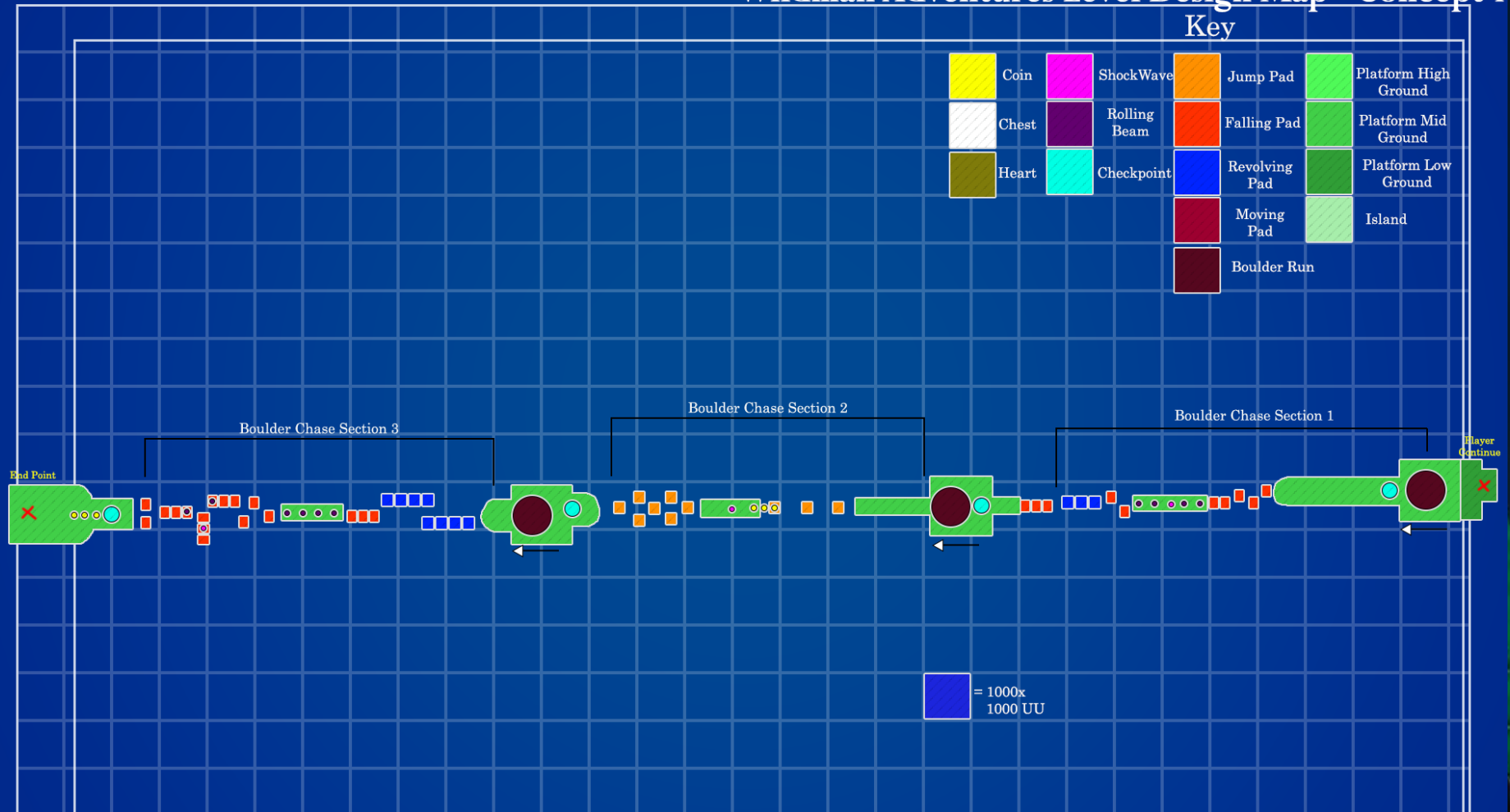
	Coin		ShockWave		Jump Pad		Platform High Ground
	Chest		Rolling Beam		Falling Pad		Platform Mid Ground
	Heart		Checkpoint		Revolving Pad		Platform Low Ground
			Moving Pad		Island		
			Boulder Run				

 = 1000x
1000 UU

 Continue Point

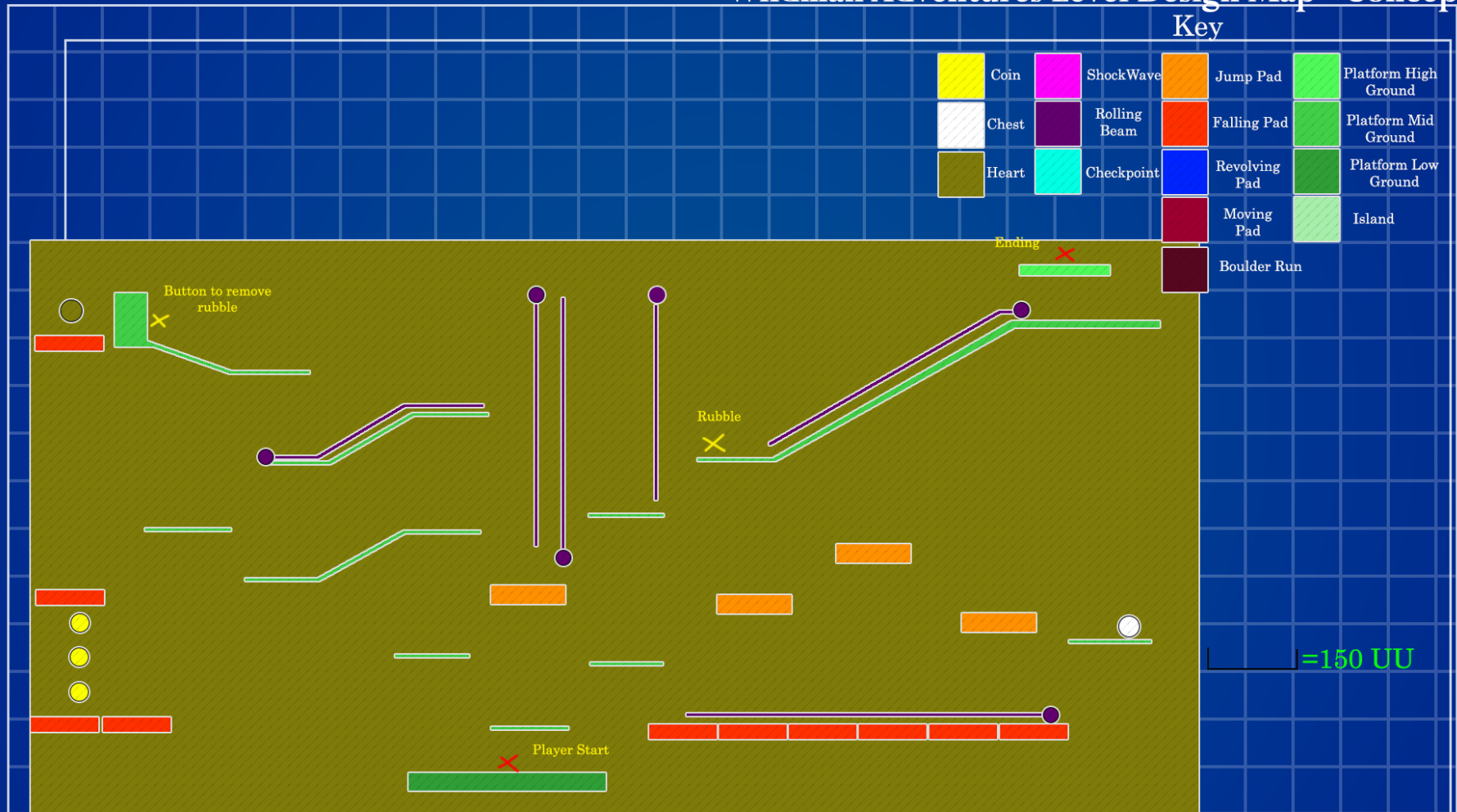
Boulder Run

Wildman Adventures Level Design Map - Concept 1



2D Puzzle Section

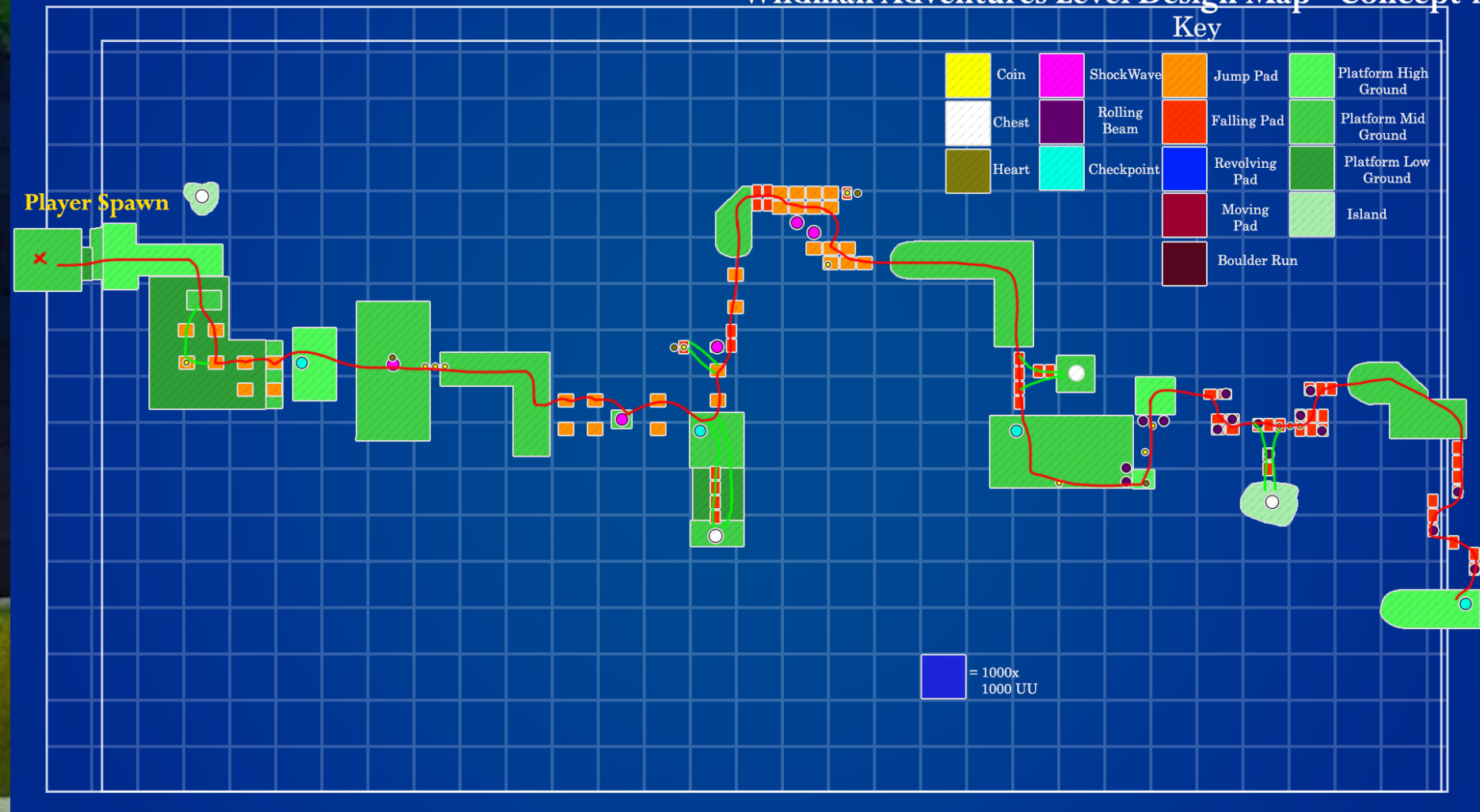
Wildman Adventures Level Design Map - Concept 1



Critical Paths

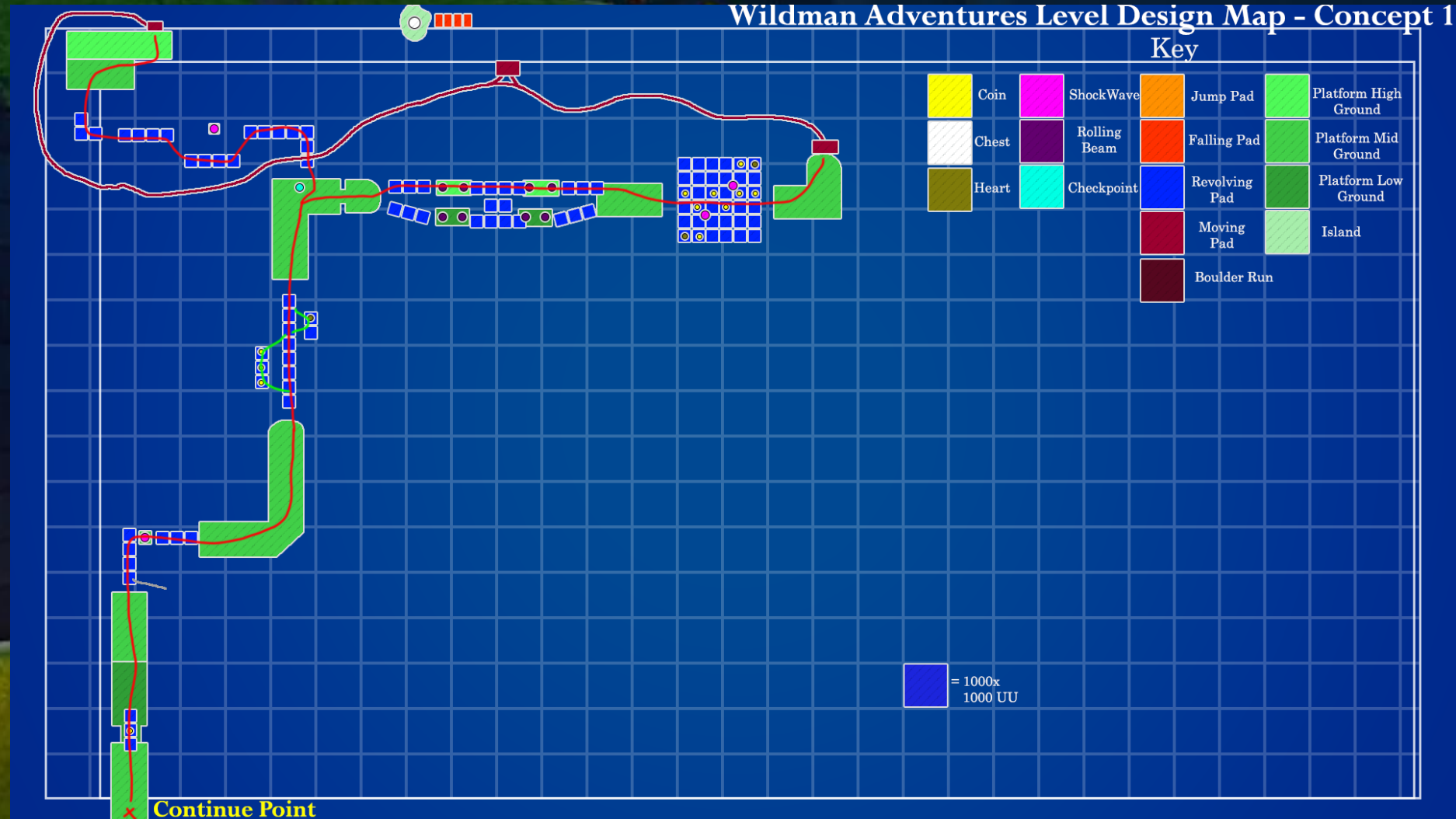
Area 1

Wildman Adventures Level Design Map - Concept 1



Area 2

Wildman Adventures Level Design Map - Concept 1



Wildman Adventures Level Design Map - Concept 1

