Design

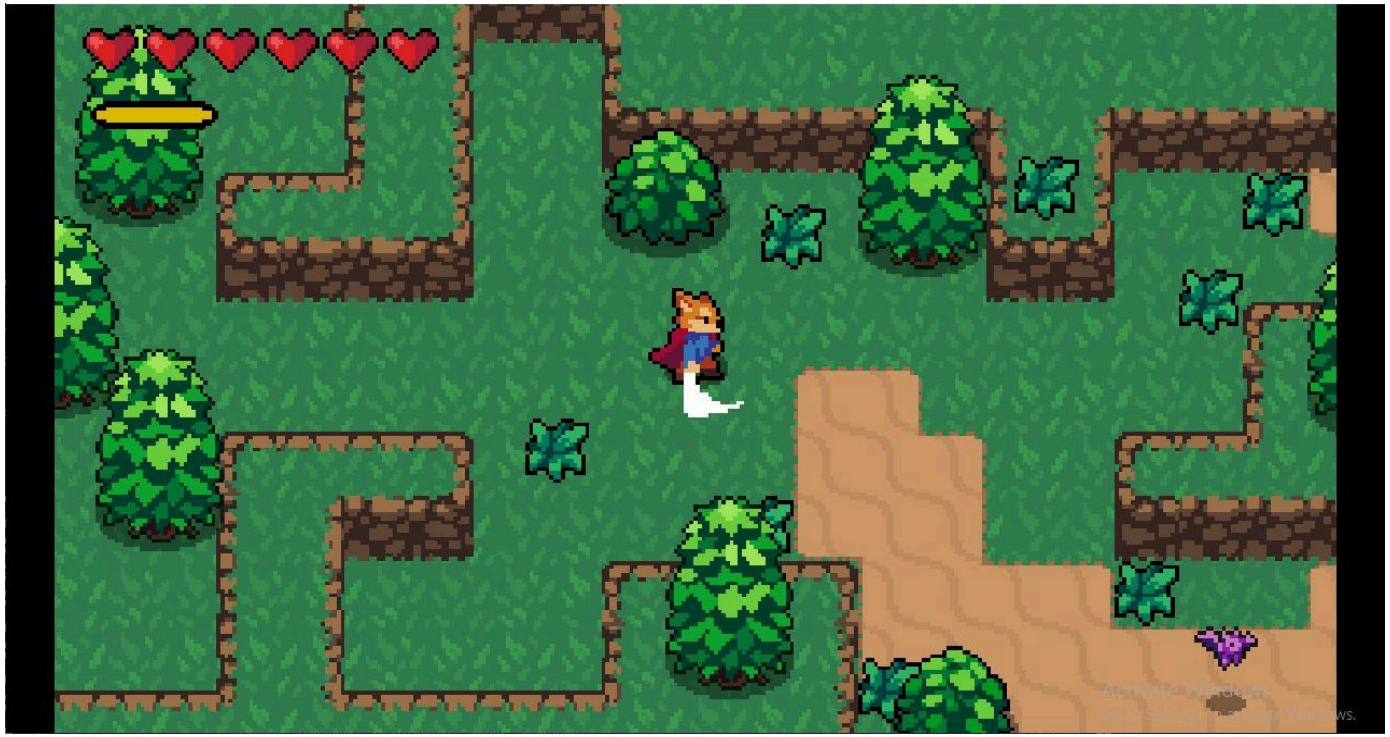
GUI Element 1: Start Screen

The client enter any key to start and sends them to the loading screen

GUI Element 2: Loading Screen

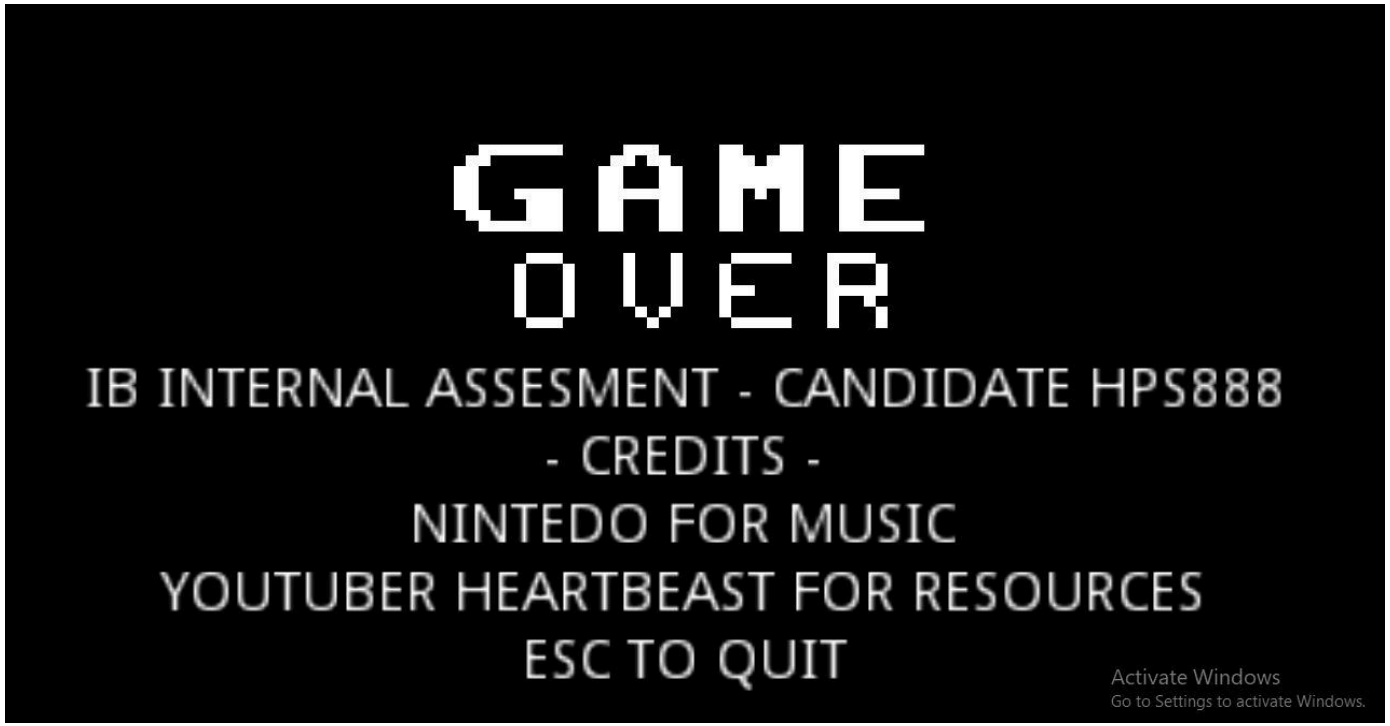
This screen is designed to give the client knowledge about the input they may take

GUI Element 3: World/Forest Scene



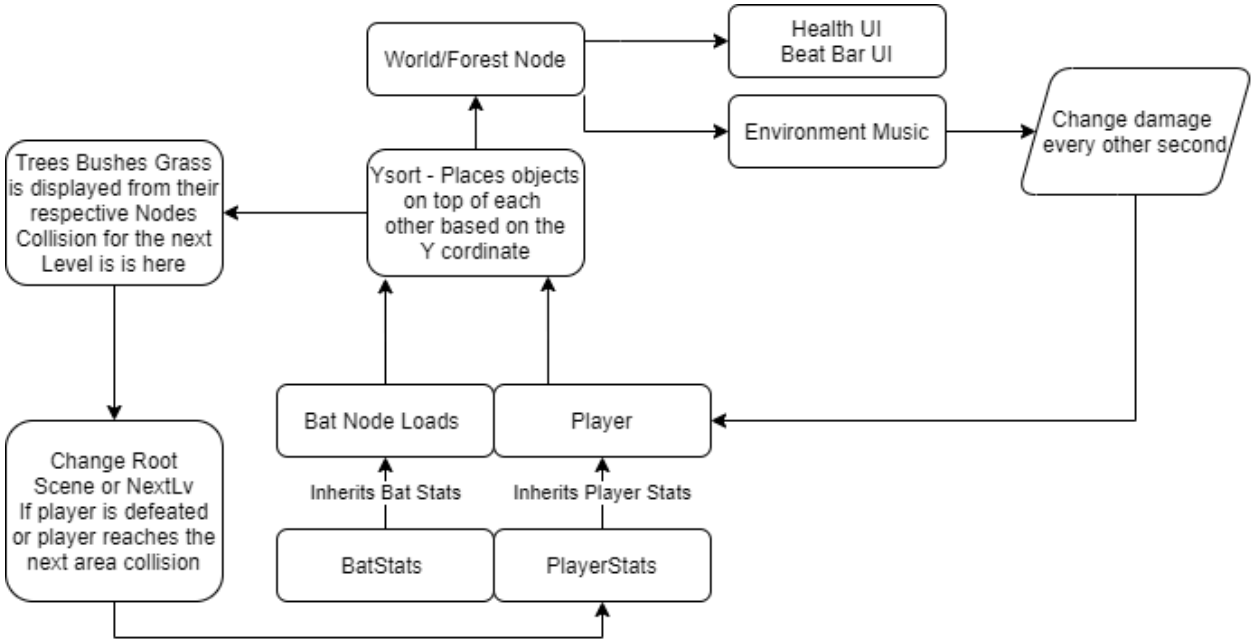
This is the environment that the player is to interact with, 2 GUI items are displayed:
Hearts - Amount of hit the Player body can take | Beat Meter - Glows when damage increase

GUI Element 4: Game/Simulation Over Scene

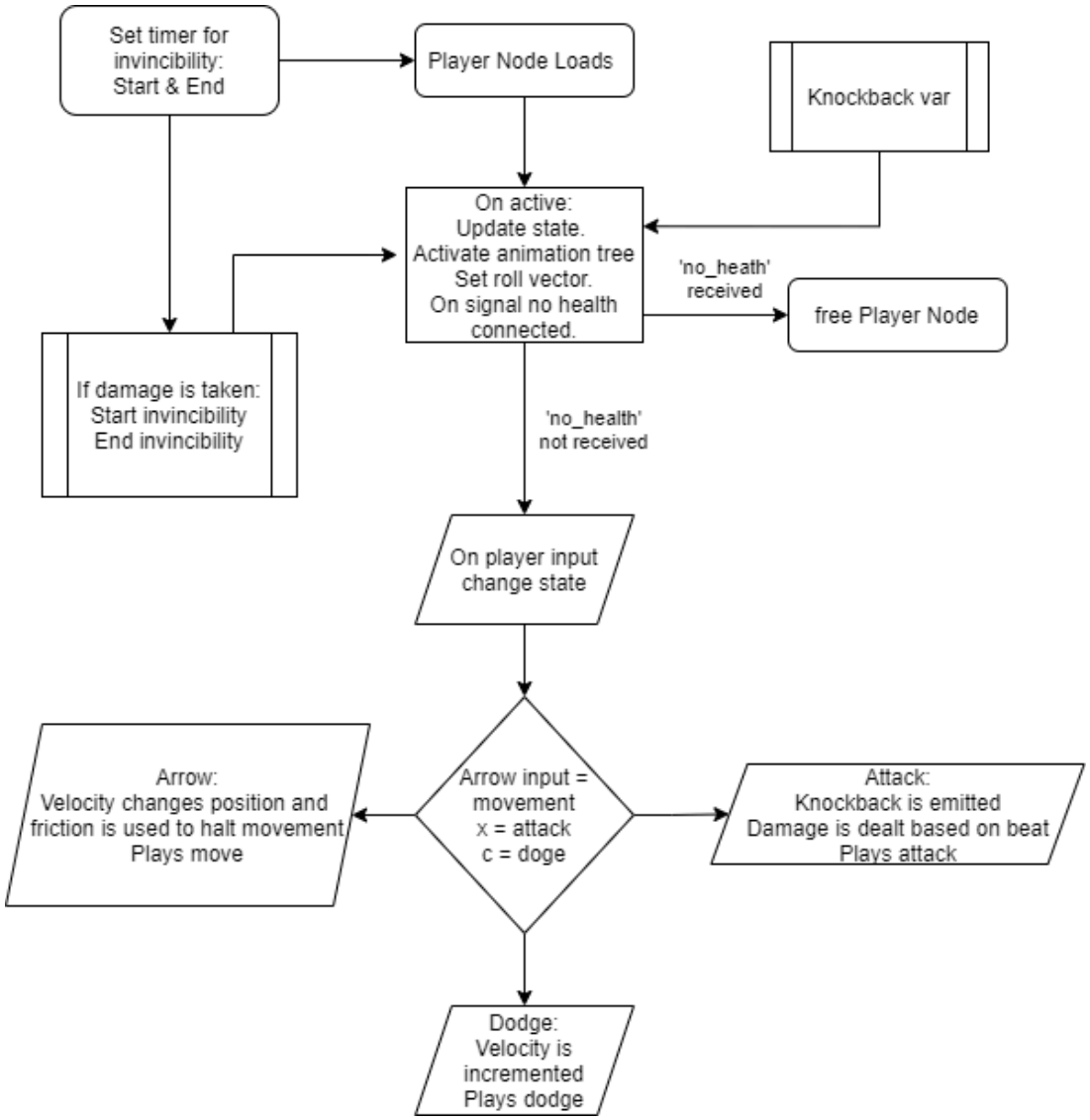


This gives the client the credits to the developer, resource provider, and a way to exit

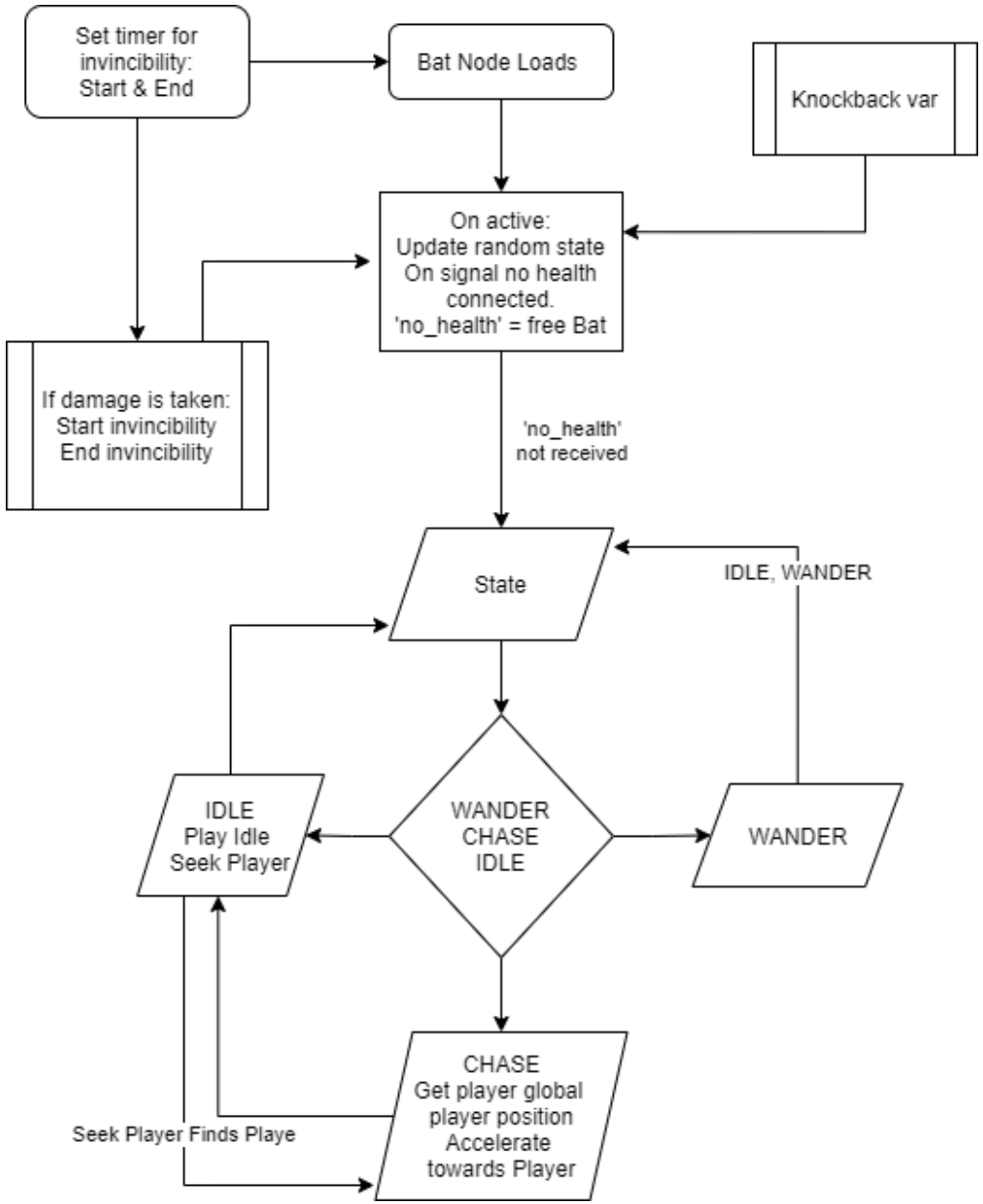
Flowchart - World



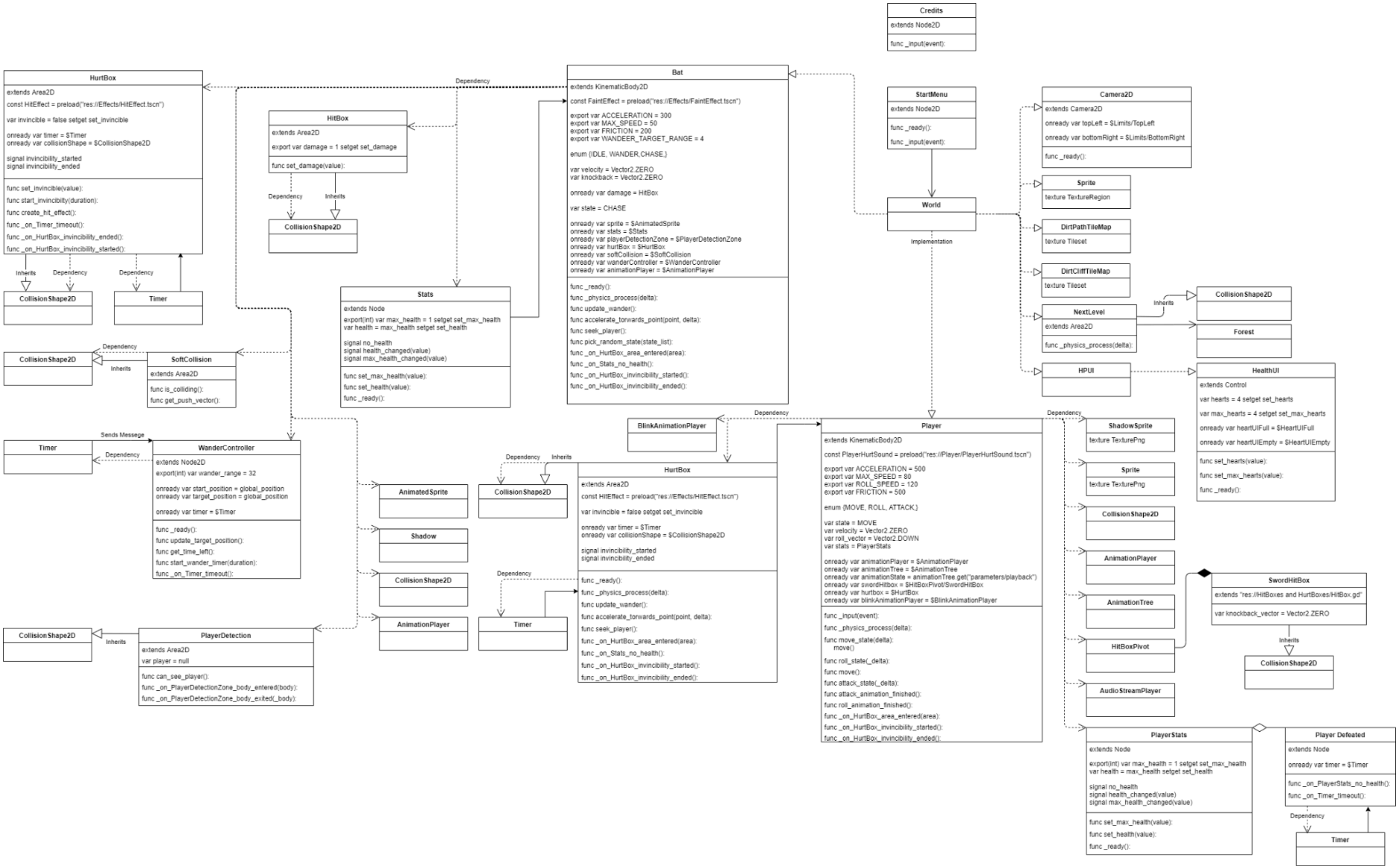
Flowchart - Player



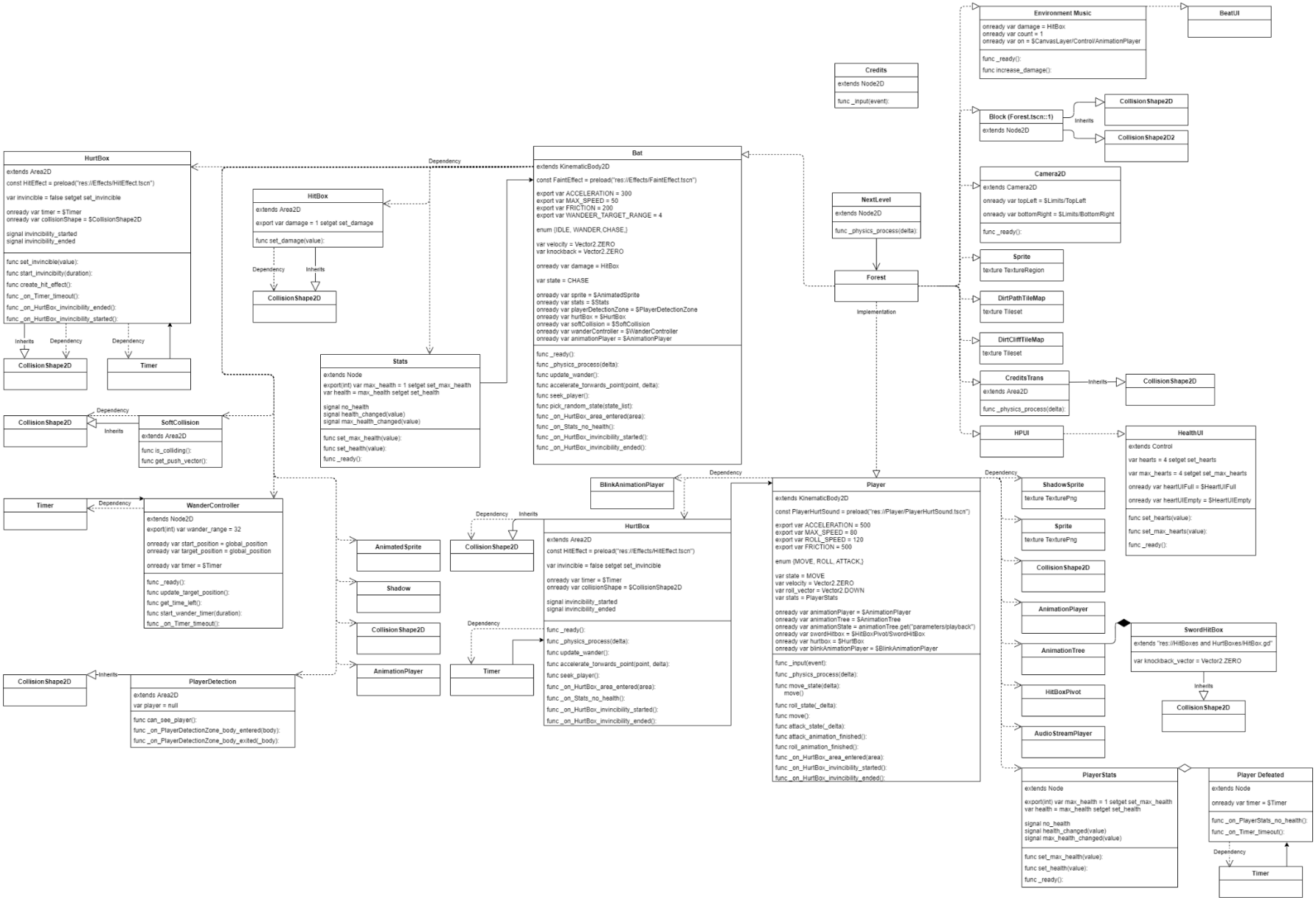
Flowchart - Bat



UML - World


Credits
extends Node2D
func _input(event):
HurtBox
extends Area2D
const HitEffect = preload("res://Effects/HitEffect.tscn")
var invincible = false setget set_invincible
onready var timer = $Timer
onready var collisionShape = $CollisionShape2D
signal invincibility_started
signal invincibility_ended
func set_invincible(value):
func start_invincibility(duration):
func create_hit_effect():
func _on_Timer_timeout():
func _on_HurtBox_invincibility_ended():
func _on_HurtBox_invincibility_started():
Dependency
Bat
extends KinematicBody2D
const FaintEffect = preload("res://Effects/FaintEffect.tscn")
export var ACCELERATION = 300
export var MAX_SPEED = 50
export var FRICTION = 200
export var WANDEER_TARGET_RANGE = 4
enum {IDLE, WANDER,CHASE,}
var velocity = Vector2.ZERO
var knockback = Vector2.ZERO
onready var damage = HitBox
var state = CHASE
onready var sprite = $AnimatedSprite
onready var stats = $Stats
onready var playerDetectionZone = $PlayerDetectionZone
onready var hurtBox = $HurtBox
onready var softCollision = $SoftCollision
onready var wanderController = $WanderController
onready var animationPlayer = $AnimationPlayer
func _ready():
func _physics_process(delta):
func update_wander():
func accelerate_towards_point(point, delta):
func seek_player():
func pick_random_state(state_list):
func _on_HurtBox_area_entered(area):
func _on_Stats_no_health():
func _on_HurtBox_invincibility_started():
func _on_HurtBox_invincibility_ended():
StartMenu
extends Node2D
func _ready():
func _input(event):
Camera2D
extends Camera2D
onready var topLeft = $Limits/TopLeft
onready var bottomRight = $Limits/BottomRight
func _ready():
Sprite
texture TextureRegion
World
Implementation
DirtPathTileMap
texture Tileset
DirtCliffTileMap
texture Tileset
NextLevel
extends Area2D
func _physics_process(delta):
Inherits
CollisionShape2D
Forest
HPUI
HealthUI
extends Control
var hearts = 4 setget set_hearts
var max_hearts = 4 setget set_max_hearts
onready var heartUIFull = $HeartUIFull
onready var heartUIEmpty = $HeartUIEmpty
func set_hearts(value):
func set_max_hearts(value):
func _ready():
HitBox
extends Area2D
export var damage = 1 setget set_damage
func set_damage(value):
Dependency
Inherits
CollisionShape2D
Stats
extends Node
export(int) var max_health = 1 setget set_max_health
var health = max_health setget set_health
signal no_health
signal health_changed(value)
signal max_health_changed(value)
func set_max_health(value):
func set_health(value):
func _ready():
CollisionShape2D
SoftCollision
extends Area2D
func is_colliding():
func get_push_vector():
Timer
Sends Message
WanderController
extends Node2D
export(int) var wander_range = 32
onready var start_position = global_position
onready var target_position = global_position
onready var timer = $Timer
func _ready():
func update_target_position():
func get_time_left():
func start_wander_timer(duration):
func _on_Timer_timeout():
AnimatedSprite
Shadow
CollisionShape2D
AnimationPlayer
BlinkAnimationPlayer
HurtBox
extends Area2D
const HitEffect = preload("res://Effects/HitEffect.tscn")
var invincible = false setget set_invincible
onready var timer = $Timer
onready var collisionShape = $CollisionShape2D
signal invincibility_started
signal invincibility_ended
func _ready():
func _physics_process(delta):
func update_wander():
func accelerate_towards_point(point, delta):
func seek_player():
func _on_HurtBox_area_entered(area):
func _on_Stats_no_health():
func _on_HurtBox_invincibility_started():
func _on_HurtBox_invincibility_ended():
Timer
CollisionShape2D
PlayerDetection
extends Area2D
var player = null
func can_see_player():
func _on_PlayerDetectionZone_body_entered(body):
func _on_PlayerDetectionZone_body_exited(_body):
Player
extends KinematicBody2D
const PlayerHurtSound = preload("res://Player/PlayerHurtSound.tscn")
export var ACCELERATION = 500
export var MAX_SPEED = 80
export var ROLL_SPEED = 120
export var FRICTION = 500
enum {MOVE, ROLL, ATTACK,}
var state = MOVE
var velocity = Vector2.ZERO
var roll_vector = Vector2.DOWN
var stats = PlayerStats
onready var animationPlayer = $AnimationPlayer
onready var animationTree = $AnimationTree
onready var animationState = animationTree.get("parameters/playback")
onready var swordHitbox = $HitBoxPivot/SwordHitBox
onready var hurtbox = $HurtBox
onready var blinkAnimationPlayer = $BlinkAnimationPlayer
func _input(event):
func _physics_process(delta):
func move_state(delta):
move()
func roll_state(_delta):
func move():
func attack_state(_delta):
func attack_animation_finished():
func roll_animation_finished():
func _on_HurtBox_area_entered(area):
func _on_HurtBox_invincibility_started():
func _on_HurtBox_invincibility_ended():
ShadowSprite
texture TexturePng
Sprite
texture TexturePng
CollisionShape2D
AnimationPlayer
AnimationTree
HitBoxPivot
AudioStreamPlayer
SwordHitBox
extends "res://HitBoxes and HurtBoxes/HitBox.gd"
var knockback_vector = Vector2.ZERO
CollisionShape2D
PlayerStats
extends Node
export(int) var max_health = 1 setget set_max_health
var health = max_health setget set_health
signal no_health
signal health_changed(value)
signal max_health_changed(value)
func set_max_health(value):
func set_health(value):
func _ready():
Player Defeated
extends Node
onready var timer = $Timer
func _on_PlayerStats_no_health():
func _on_Timer_timeout():
Timer

UML - Forest


Environment Music
onready var damage = HitBox
onready var count = 1
onready var on = $CanvasLayer/Control/AnimationPlayer
func _ready():
func increase_damage():
BeatUI
Credits
extends Node2D
func _input(event):
Block (Forest.tscn::1)
extends Node2D
Inherits
CollisionShape2D
CollisionShape2D2
Camera2D
extends Camera2D
onready var topLeft = $Limits/TopLeft
onready var bottomRight = $Limits/BottomRight
func _ready():
HurtBox
extends Area2D
const HitEffect = preload("res://Effects/HitEffect.tscn")
var invincible = false setget set_invincible
onready var timer = $Timer
onready var collisionShape = $CollisionShape2D
signal invincibility_started
signal invincibility_ended
func set_invincible(value):
func start_invincibilty(duration):
func create_hit_effect():
func _on_Timer_timeout():
func _on_HurtBox_invincibility_ended():
func _on_HurtBox_invincibility_started():
Dependency
Inherits
CollisionShape2D
Timer
HitBox
extends Area2D
export var damage = 1 setget set_damage
func set_damage(value):
Dependency
Inherits
CollisionShape2D
Bat
extends KinematicBody2D
const FaintEffect = preload("res://Effects/FaintEffect.tscn")
export var ACCELERATION = 300
export var MAX_SPEED = 50
export var FRICTION = 200
export var WANDEER_TARGET_RANGE = 4
enum {IDLE, WANDER,CHASE,}
var velocity = Vector2.ZERO
var knockback = Vector2.ZERO
onready var damage = HitBox
var state = CHASE
onready var sprite = $AnimatedSprite
onready var stats = $Stats
onready var playerDetectionZone = $PlayerDetectionZone
onready var hurtBox = $HurtBox
onready var softCollision = $SoftCollision
onready var wanderController = $WanderController
onready var animationPlayer = $AnimationPlayer
func _ready():
func _physics_process(delta):
func update_wander():
func accelerate_torwards_point(point, delta):
func seek_player():
func pick_random_state(state_list):
func _on_HurtBox_area_entered(area):
func _on_Stats_no_health():
func _on_HurtBox_invincibility_started():
func _on_HurtBox_invincibility_ended():
Stats
extends Node
export(int) var max_health = 1 setget set_max_health
var health = max_health setget set_health
signal no_health
signal health_changed(value)
signal max_health_changed(value)
func set_max_health(value):
func set_health(value):
func _ready():
NextLevel
extends Node2D
func _physics_process(delta):
Forest
Implementation
Sprite
texture TextureRegion
DirtPathTileMap
texture Tileset
DirtCliffTileMap
texture Tileset
CreditsTrans
extends Area2D
func _physics_process(delta):
Inherits
CollisionShape2D
HPUI
HealthUI
extends Control
var hearts = 4 setget set_hearts
var max_hearts = 4 setget set_max_hearts
onready var heartUIFull = $HeartUIFull
onready var heartUIEmpty = $HeartUIEmpty
func set_hearts(value):
func set_max_hearts(value):
func _ready():
CollisionShape2D
SoftCollision
extends Area2D
func is_colliding():
func get_push_vector():
Dependency
Inherits
Timer
WanderController
extends Node2D
export(int) var wander_range = 32
onready var start_position = global_position
onready var target_position = global_position
onready var timer = $Timer
func _ready():
func update_target_position():
func get_time_left():
func start_wander_timer(duration):
func _on_Timer_timeout():
Dependency
AnimatedSprite
Shadow
CollisionShape2D
AnimationPlayer
BlinkAnimationPlayer
HurtBox
extends Area2D
const HitEffect = preload("res://Effects/HitEffect.tscn")
var invincible = false setget set_invincible
onready var timer = $Timer
onready var collisionShape = $CollisionShape2D
signal invincibility_started
signal invincibility_ended
func _ready():
func _physics_process(delta):
func update_wander():
func accelerate_torwards_point(point, delta):
func seek_player():
func _on_HurtBox_area_entered(area):
func _on_Stats_no_health():
func _on_HurtBox_invincibility_started():
func _on_HurtBox_invincibility_ended():
Timer
CollisionShape2D
PlayerDetection
extends Area2D
var player = null
func can_see_player():
func _on_PlayerDetectionZone_body_entered(body):
func _on_PlayerDetectionZone_body_exited(_body):
Player
extends KinematicBody2D
const PlayerHurtSound = preload("res://Player/PlayerHurtSound.tscn")
export var ACCELERATION = 500
export var MAX_SPEED = 80
export var ROLL_SPEED = 120
export var FRICTION = 500
enum {MOVE, ROLL, ATTACK,}
var state = MOVE
var velocity = Vector2.ZERO
var roll_vector = Vector2.DOWN
var stats = PlayerStats
onready var animationPlayer = $AnimationPlayer
onready var animationTree = $AnimationTree
onready var animationState = animationTree.get("parameters/playback")
onready var swordHitbox = $HitBoxPivot/SwordHitBox
onready var hurtbox = $HurtBox
onready var blinkAnimationPlayer = $BlinkAnimationPlayer
func _input(event):
func _physics_process(delta):
func move_state(delta):
move()
func roll_state(_delta):
func move():
func attack_state(_delta):
func attack_animation_finished():
func roll_animation_finished():
func _on_HurtBox_area_entered(area):
func _on_HurtBox_invincibility_started():
func _on_HurtBox_invincibility_ended():
ShadowSprite
texture TexturePng
Sprite
texture TexturePng
CollisionShape2D
AnimationPlayer
AnimationTree
HitBoxPivot
AudioStreamPlayer
SwordHitBox
extends "res://HitBoxes and HurtBoxes/HitBox.gd"
var knockback_vector = Vector2.ZERO
Inherits
CollisionShape2D
PlayerStats
extends Node
export(int) var max_health = 1 setget set_max_health
var health = max_health setget set_health
signal no_health
signal health_changed(value)
signal max_health_changed(value)
func set_max_health(value):
func set_health(value):
func _ready():
Player Defeated
extends Node
onready var timer = $Timer
func _on_PlayerStats_no_health():
func _on_Timer_timeout():
Dependency
Timer