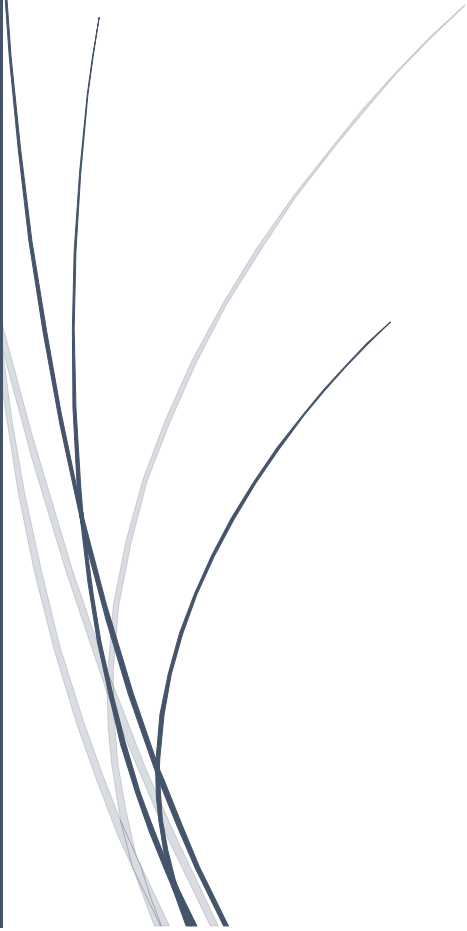




9/7/2020

Game Script Design Document



Contents

All the different mechanics in the game.....	2
General mechanics:	2
The history book	2
Gameplay	2
Level-based mechanics:	2
Gameplay (in chapters)	2
For each mechanic, identify and outline classes and scripts (give fields, method names, script/class names) that would be necessary for that mechanic to work:	4
Central scripts	4
The history book.....	4
Gameplay	4
Cameras	5
NPC's.....	5
Sound	6
Discuss how mechanics scaffold and build on one another	7
Diagram:.....	7
Building on:	7
The history book	7
General gameplay mechanics:	7
Level-based mechanics	8

All the different mechanics in the game

With the design of the game being based in a history book with each page have its own unique gameplay level focusing on the lifecycle of the black rhino, the games mechanics are divided into to sections, general mechanics and level based mechanics. These two sections will be explained in more detail below but in summary, general mechanics are mechanics that can be found all across the game that the player will interact with on multiple occasions and level based mechanics are mechanics that the player will interact will only in certain gameplay levels within the game.

General mechanics:

The history book

- Page turning
 - With the setting of a game taking place in a history book, it is important that there is a smooth navigation mechanic that the player can utilize to smoothly transition between the different pages of the book (more specifically the different gameplay levels), this navigation mechanic is the page turning
 - The player will be able to hold the edge of a page and drag it to the opposite side of the screen, which will in turn change the page of the book the player is currently on
 - If the player holds the edge of the left page and drags to the right of the screen, they will be navigating back in the book to a previous page
 - If the player holds the edge of the right page and drags to the left of the screen, they will be navigating forward in the book to the next page.

Gameplay

- Movement
 - Throughout the different gameplay levels, the player will have the ability to move their character (the rhino) within the constraints of the levels 3D space
 - Although the movement may vary across different gameplay levels (mainly in speed) it will always be controlled with W, A, S and D
- Camera controls
 - Throughout the different gameplay levels, the player will be able to change the position of their 3rd person camera, rotating it around their character
 - Although the player can rotate the camera, it will always stay a fixed distance away from the character

Level-based mechanics:

Gameplay (in chapters)

- Chapter 1: First Steps (level 1)
 - Standing up
 - With the levels focus on the main character's first steps, the player will be required to make the rhino stand up when it falls due its clumsy nature after birth
 - If the rhino is currently lying on the ground, the player will need to tap the SPACE button in quick procession to gradually lift the baby rhino onto its feet
- Chapter 2: Eating and Drinking (level 2)
 - Eating and drinking
 - With the levels focus on the main character's hunger and thirst, the player will be required to:

- Find sources of food (plants that rhinos eat) and water
 - Once a source of food or water has been found, the player will be required to click E in order eat the source of food or drink from the source of water
 - Thirst and malnutrition mechanic
 - With a focus on eating and drinking, the player will be driven to do so with the thirst and malnutrition mechanic within the gameplay level
 - Although the time frame of this mechanic is not realistic, it is implemented into the gameplay level to drive the gameplay
 - Visual indicators (in the form of UI) will indicate, to the player, the state of the rhino's thirst and hunger. If one of these indicators reach 0 the player will ultimately fail the gameplay level and be required to restart
- Chapter 4: The Game Ranger (level 4)
 - Inspecting
 - With the gameplay level's focus on creating an emphasis that the rhino is very unfamiliar with humans, the play will be required to inspect the objects that the game ranger brings along with him when visiting the rhino for the first time, The player will be required to inspect the following objects:
 - The game rangers vehicle
 - His binoculars
 - His hat
 - In order to inspect an object, the player will need to be in a specific range of the object. Once in the specific range of an object the player will be required to click E to inspect it
 - Interacting
 - Once the player has successfully inspected all of the game rangers items, they will then be required to interact with the game ranger.
 - Interacting works very similar to inspecting, the player will be required to get in a specific range of the game ranger. Once in the specific range the player can click E to initiate the interaction.
- Chapter 5: Trying to escape
 - Detection
 - With the levels focus on a poacher trying to find the rhino, the player will need to maneuver around the 3D space in order to not get detected and found by the poacher
 - The poachers detection range will be visually portrayed as a large circle around him. If the player (rhino) is caught in this circle the poacher will start becoming more aware of your position and start moving towards you
 - To emphasize on an inevitable fate, the poachers circle will constantly keep growing, giving him more detection range. Eventually the player (rhino) will not be able to evade detection anymore

For each mechanic, identify and outline classes and scripts (give fields, method names, script/class names) that would be necessary for that mechanic to work:

Central scripts

PlayManager
--playerName:string
-currentChapter: int
-Start()
+Save()
+Load()
+StartGame(int)

The history book

HistoryBook_Controller
-pageNumber: int
-pCanvasController: PageCanvas_Controller
+playManager: PlayManager
-Start()
-FixedUpdate()
-nextPage()
-previousPage()
+getPageNumber()

HBCanvas_Controller
-pageCanvases: List<Canvas>
-pCanvasController: PageCanvas_Controller
-Start()
-FixedUpdate()
+assignNextPageUI(int)
+assignPreviousPageUI (int)

Gameplay

Player_Controller
+moveSpeed: float
+canMoveBackwards: boolean
+canMove: boolean

+playManager: PlayManager
-Start()
-FixedUpdate()
-Base_Movement()
-Chapter1_Movement()
-Interaction()
-OnTriggerEnter (Collider)

Cameras

GameCameras_Manager
+gameCamera: List<Camera>
+playManager: PlayManager
-Start()
-FixedUpdate()
+getPlayerCamera(int)

PlayerCamera_Controller
+rotationSpeed: float
+lockedYAxis: boolean
+canRotate: boolean
+playerCamera: Camera
-Start()
-FixedUpdate()
-cameraRotation()
-findPlayerCamera()

NPC's

NPC_Manager
+playManager: PlayManager
+mrIsActive: bool
+poacherIsActive: bool
-Start()
-FixedUpdate()

MotherRhino_Controller
+mrAgent: NavMeshAgent
+moveSpeed: float

+movePoints: List<GameObject>
-atDestination: bool
+npcManager: NPC_Manager
-Start()
-FixedUpdate()
-Movement()

Poacher_Controller
+poacherAgent: NavMeshAgent
+moveSpeed: float
+startingDetectionRadius: float
+npcManager: NPC_Manager
-Start()
-FixedUpdate()
-Movement()
-Patrol()
-Aware()
-Chase()

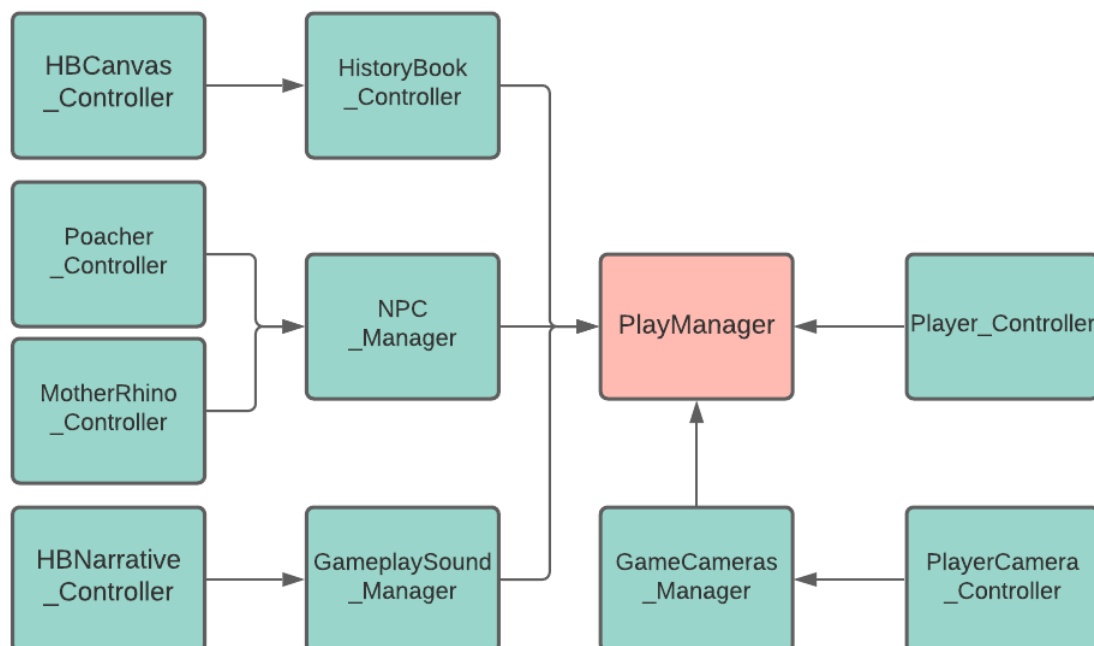
Sound

GameplaySound_Manager
+gameSounds: List<AudioClip>
+pManager: playManager
-Start()
+playSound(int)

HBNarrative_Controller
-pageNarrative: List<AudioClip>
+gsManager: GameplaySound_Manager
-Start()
-FixedUpdate()
+playPageNarrative(int)

Discuss how mechanics scaffold and build on one another

Diagram:



The diagram above represents the structure of the game's scripts. It displays which scripts communicate with one another and which scripts control other scripts (Manager scripts). The design of the game's architecture is focused around one central manager (PlayManager), providing a central point of communication between all of the scripts within the game, allowing for the interaction between multiple mechanics and systems to create a smooth and responsive gameplay experience for the player.

Building on:

The history book

The history book is interacted with throughout the game, turning pages to access the next chapters of gameplay. Depending on what page (chapter) the player is on, they will experience different levels of gameplay-many with their own mechanics. When a page is turned, information is sent to the scripts containing the mechanics for that chapter's gameplay, activating the mechanic/s and initiating the gameplay sequence.

General gameplay mechanics:

Throughout each chapter's gameplay, there are two mechanics that are always used. These mechanics, movement and camera control, are the foundation for majority of the other mechanics within the chapters-often being required to be used in order for the player to access the other mechanics in each chapter.

- Chapter 1:
 - In order for the rhino to take its first steps, the player is required to engage with the movement mechanic aiming to keep the baby rhino on its feet. Although the first steps mechanic does vary a bit from the general movement mechanic, its base inherits the core body of the general movement mechanic.

- Chapter 2:
 - In order for the rhino to initiate the drink or eat mechanic, the player must first use the movement mechanic to get to sources of food or water
- Chapter 3:
 - The entirety of chapter 3's gameplay is driven through the movement mechanic. The player must use it in order to move across the land to find a new place for the rhino to start its solitary life
- Chapter 4:
 - Very similar to chapter 2, the player can only access the chapters unique mechanics with the use of the movement mechanic. In order for the player to get to the objects and game ranger they need to interact and expect, they must use the movement mechanic to maneuver around the game world in order to do so
- Chapter 5:
 - The evade mechanic is built upon the movement mechanic, requiring the player to utilize the movement mechanic in order to evade and escape the poacher

Throughout all of the chapters gameplay, the player is able to access the camera controls mechanics to observe their surroundings in the game world.

Level-based mechanics

Many chapters within the game have their own unique mechanics. Due to the mechanics being unique to the chapter there is no building on from one another (unique chapter mechanics). Although there are some chapters where unique chapter mechanics build upon the general gameplay mechanics within the game (explained above), there is little to no communication between the gameplay of one chapter's unique mechanics to another's.