

UnityDialogueUI : MonoBehavior

Public void TransistionToState(IState newState)

Public dialogueText;
Public affectionText;
Public Buttons (1,2)
Private dialogueController

Void start()
// set up buttons, create dynamic
// types of sonic/shadow
Void ShowDialogue()
Void OnResponse(int choice)

SonicDialogue: HedgehogDialogue

Public AffectionManager affectionManager
Public static bool lockoutSonic
Public string game
Public string playerName;

Public List<string> sonicLines
Public List<string[]> playerResponses
Public override void ProcessChoice()
Public void EndConversation()
Public override string[]
GetCurrentResponses()
Public List<string> GetSonicLines
Public override bool
IsConversationFinished()
Public static bool CheckSonicLockout
Public void UpdateDialogueAfterMinigame()

ShadowDialogue: HedgehogDialogue

Public AffectionManager affectionManager
Public string playerName;
Public static bool lockoutShadow
Private bool isFinished
Public string game

Public List<string> shadowLines
Public List<string[]> playerResponses
Public override void ProcessChoice()
Public void EndConversation()
Public override string[]
GetCurrentResponses()
Public List<string> GetShadowLines
Public override bool
IsConversationFinished()
Public static bool CheckShadowLockout
Public void UpdateDialogueAfterMinigame()

SonicNormallState: IState

Private SonicDialogue dialogue;

Public void EnterState()
Public void ProcessChoice(int)

SonicMiniGameState: IState

Private SonicDialogue dialogue;

Public void EnterState()
Public void ProcessChoice(int)

SonicLockoutlState: IState

Private SonicDialogue dialogue;

Public void EnterState()
Public void ProcessChoice(int)

ShadowLockoutlState: IState

private ShadowDialogue dialogue;

Public void EnterState()
Public int ProcessChoice(int)

ShadowMiniGameIState: IState

private ShadowDialogue dialogue;

Public void EnterState()
Public void ProcessChoice(int)

ShadowNormalState: IState

private ShadowDialogue dialogue;

Public void EnterState()
Public void ProcessChoice(int)