ROAD TRIP

A dark choice-based adventure game told from the perspective of a psychologically damaged 14-year-old girl.

Game Design Document

Prototype 1 – Group 1

Devan Gray (1615858)
Leia French (1600474)
Victoria Bench (1611349)
Marco Prinsloo (1363650)
Keaton De Jager (1636214)

Contents

1. GAME OVERVIEW

1.1. Game Concept

The player plays as a fourteen-year-old girl who has been kidnapped. The game is a series of decisions inspired by games such as *Until Dawn* [6] and *Detroit: Become Human* [5]. Each decision will lead the player down a web of narrative adventure ultimately leading to either being restrained further, successfully escaping, or killing the men who shot her father. The idea behind this interactive narrative is to serve as the first level of a mysterious and dark story told from the perspective of a psychologically damaged teenage girl. For this reason, the art style chosen for the game is meant to appear childish and innocent. This will also enhance the uncomfortable feelings provoked by the subject matter. The art style is inspired by Nyamnam's *Tengami* [2].

1.2. Key Features

The game will combine a dark and mysterious narrative with a childish and innocent aesthetic to create a sense of discomfort in the player. The player will progress through the game using a choice-based butterfly-effect style of storytelling.

1.3. Genre

Psychological Thriller

1.4. Target Audience

The game is intended to be played by people between the ages of 16 and 24 who enjoy narrative games. Content may be more traumatising for younger audiences. Simultaneously, the game will relate more to people who are closer to the character's age group – given that the protagonist is 14 years old.

1.5. Look and Feel

For the purposes of this prototype, the assets were designed practically to demonstrate how the game would work in three dimensions from a first-person perspective. The desired aesthetic for the next iteration(s) of the game is to mimic a pop-up up book similar to Nyamnam's *Tengami*, in order to further convey the theme of storytelling through a more innocent perspective [2]. In the prototype, however, the aesthetics utilize previously rigged three dimensional assets that may not fully fit the desired aesthetic but are there for demonstrative purposes.

2. MECHANICS, DYNAMICS AND AESTHETICS

For the following breakdown of the mechanics, dynamics and aesthetics; the MDA framework is used as the basis of the analysis.

2.1. Mechanics

2.1.1. Making Choices

- The player is presented with options of how to act.
- The player must choose one of those options that determines how the game will continue.
- The options are dialogue based as well as action based (What does the character say, or What does the character do?)

2.1.2. Looking Around

- The player can visually explore the game space.
- Looking around the space determines when options are presented to the player.
- The player will never be able to physically move within the game but instead only be able to rotate their head to look around the environment. This enables tension and immersion (given that the protagonist is restrained).

2.1.3. Quick Time Events

- Certain decisions must be made quickly.
- The player may have to execute instructions on the screen as they appear. (such as in God of War 3 during boss fights and other similar quick time events)

2.1.4. *Controller Input Combinations*

- The player will partake in minigames when they choose certain options
- The minigames will be made up of analog stick rotations and quick time combinations
- The combinations will try to make sense with the action being done.

2.2. *Dynamics*

2.2.1. *Prediction*

- The mechanic of choice brings about the dynamic of prediction
- Players can attempt to justify their actions by predicting the outcomes.
- The player must be aware of the consequences of their actions, by predicting the impact of said consequences.
- Predicting or anticipating quick time events becomes the difference between experienced and inexperienced players. This creates a sense of progression for the player as they become better at predicting the system.

2.2.2. *Evasion*

- Avoiding the detection of your captors creates tension
- Successful evasion creates a sense of accomplishment.
- This is both enhanced by the cost of failure in the narrative.

2.2.3. *Spatial Reasoning*

- The player must be aware of their surroundings and the objects available to them.
- Choices are only available if the player knows what is around them.
- Tools and solutions to the puzzles rely on knowing what is available to the player.

2.2.4. *Survival*

- This is the main goal of the character being controlled by the player.
- This is enhanced by making the cost of failure quite high to deter players from seeing failure as an annoying set back. Failure as a player means you let the character die.

2.2.5. *Skill Challenges*

- The introduction of the minigame mechanic allows the player to improve a skill
- Improving a skill makes the player feel like progress is being made, even if the player is failing the mission objectives. They still feel improvement.

2.3. *Aesthetics*

The core aesthetics that describe the experience to be designed.

2.3.1. *Narrative*

- An intense, serious, and dark plot makes the narrative experience engaging and creates a sense of intrigue.
- The dark aspects of the story enhance the dynamics mentioned above by putting weight onto the decisions the player makes.

2.3.2. *Sensation*

- The visual aesthetic of the game is meant to be cute and innocent.
- The artstyle leads to some beautiful scenery.
- The combination of specifically designed audio and childish art styles enhances the narrative experience and makes the player experience discomfort when contemplating the plot of the game in that context.

2.3.3. *Exploration*

- The player is driven to the game by the need to know more.
- Discovery (Exploration) of narrative is the driving force behind the gameplay loop.
- The art style and narrative in tandem drives the player to see more of the game while learning more about the story.

3. GAMEPLAY

3.1. Gameplay

3.1.1. *Game Progression*

The game is played via a controller. The player experiences the game-world from a first-person point of view and has control over their view (though not their position – adding to tension). They can pick up and use objects within their reach. This enables gameplay progression because as players pick up and use certain objects, that reach will extend – for example, removing one's handcuffs will enable players to reach objects further away from them. In turn, this enables players to reach more items, resulting in progression.

3.1.2. *Puzzle Structure*

The key puzzle-solving element of the game lies in locating objects around the car that are within your reach and, more significantly, deciding which of them is best suited for the situation at hand. These objects will enable you to progress through the game both mechanically and narratively as your reach extends.

3.1.3. *Objectives*

The main objective of the game is survival.
This can be broken up into smaller sub-objectives:

- Free your hands of the hand cuffs.
- Find a way to either stop the car or get out of it while its moving.

3.1.4. *Play Flow*

The game keeps the player balanced between boredom and anxiety. The tense music, dark lighting and multitude of choices all add to the tension of the game, keeping the player from becoming bored. However, the player is given as much time and space as they need to make their decisions, ensuring that they do not become too anxious either. [3]

3.2. Game Difficulty

The game does not have options for the player to choose how difficult the game is. For the prototype, the game is difficult as it is. However, should the idea be developed further, the difficulty would ideally be dynamic. In this case, if the player struggles with events of a certain type, then those events would become less frequent or easier to beat.

3.3. Replaying and Saving

The gameplay loop is driven by the multiple branches of narrative. The player can play through the prototype to the end but may wish to know what would have happened if they had made a different choice somewhere along the way. This means that the number of endings obviously available to the player determines the effectiveness of the gameplay loop. The player must see that their choices affect the outcome.

4. STORY, SETTING AND CHARACTER

4.1. Synopsis of story

You are woken by the sound of voices coming from in front of you. The throbbing in your head makes it difficult to move too fast. Looking up you see a long and winding road. The driver's seat occupied by a rather ordinary man, except for the tattoo on his left shoulder. To his left, another man. This one much larger and more terrifying than the driver. You're sitting in the middle of the back seat of an old Cadillac. You pass in and out of consciousness for a few minutes before the realization kicks in.

Where are your parents?

Who are these people?

Where are they taking you?

You try and move but a combination of handcuffs and stiff muscles holds you in place. You overhear the men discussing something. A few phrases stand out.

"Hope the boss approves. She's not exactly what he asked for."

"... will enjoy killing her."

"Why'd you have to go and shoot the dad."

The last one shakes every bit of drowsiness from your body. You need to escape, by any means necessary. First thing is first, get these handcuffs off.

4.2. Story and Narrative

4.2.1. Characters:

Three characters are directly involved in the prototype. Namely, Nevil and Milo, who are the bad guys and Julianna who is the player character. Another character is mentioned but not important, and that is the boss of the bad guys.

4.2.2. The story points are as follows:

- The player character has been kidnapped by two thugs.
- She is being taken to their boss.
- She was not the original target of the kidnapping.
- The original target is a mystery and would become a major plot point later in the story.
- The player character's father has been killed by one of the thugs.
- The player character can escape from the car by getting out of the handcuffs and either killing the driver or breaking the window and jumping out of the car.

4.2.3. Game World Area

The game takes place solely inside the car of your captors.

5. LEVELS

The prototype will contain a single level. This level takes place inside the car of your captors and is focussed on escaping said car.

This prototype will not contain any tutorial – but instead will ease the player into gameplay with a slow progression of game objectives that will happen one by one.

6. GAME ART AND SOUND

6.1. Art Assets and Intended style

The boxiness of the car moves closer to the paper-based environment that we strive to achieve. The person objects don't match this aesthetic, centrally because they were previously made and recycled out of convenience. The seats, dashboard, tools, trees, and handcuffs were designed to move toward the folded paper aesthetic, making use of flatter surfaces and colours, but still maintain some smoothness and rounded edges in their form so as to match the person models and not completely confuse the player. In the next iterations, proper character and asset design would be done to make it more cohesive overall.

Figure 1: Inspiration for the car/overall aesthetic [7]

Figure 2: Light and style inspiration for prototype [1].

6.2. Text Assets

OpenSansSemibold and MostlyMono are to be used as the fonts for the text used in the game.

6.3. Sound Assets

To fit to the overall themes of the game it was decided to go for a music theme that sounds eerie and innocent. To do this a minor key was used with the piano and violin as the instruments. The music theme was implemented by deciding on a simple two chord, chord proregression and then playing around with an initial piano melody pattern. Once this was done, a violin accompaniment was created. Work was then done on a varied piano melody, and violin accompaniment patterns, which were then arranged into a full loop able song. Some crackling effects were then added and a plugin to make the song sound like an older record song was used. Finally, the equalizer and levels of the instruments were tweaked, a volume fade in at the beginning and a volume fade out at the end of the song was both added.

7. INTERFACE

7.1. Visual System

The screen is very simple with little to no UI visible. The only UI on the screen is the choices menu, subtitles for when dialogue occurs and the instructions when minigames are active. The camera is first person and 16x9. Instructions are largely given visually rather than textually – for example, images of the Playstation control buttons will appear when making choices or playing minigames.

7.2. Control System

The game is entirely controlled with a controller. The controls for this prototype are not customizable.

Figure 3: Controls layout for the game [4].

8. TECHNICAL

8.1. Development hardware and software

8.1.1. Hardware

All members of the group will be using Dell Inspiron 15 7577 laptops for development with the following specifications:
- Intel 7th Generation Kabylake Core i7-7700HQ processors.
- 16GB of DDR4 RAM.
- 256GB ultra-fast SSD.
- 1TB hard drive.
- NVIDIA GeForce GTX 1060 6GB GDDR5 graphics cards.

8.1.2. Software used

Unity (Version 2018.4.4f1 (64-bit)) as the Game Engine for development
Unity has been chosen as the game engine used for the development of this project due to all group members being experienced in using the software.

GitHub
GitHub allows for easy integration and sharing between all members of the team. It also allows the team to assess what has been implemented and what hasn't at each stage of the project. GitHub also allows us to roll back to an earlier stage of the project, should anything go wrong.

SourceTree
The GUI application that the group will use to work with git to manage and handle the main repository used to produce this prototype.

Visual Studio Code
Visual studio code provides syntax highlighting, intelligent code completion both which will benefit our programmers in being more efficient. Visual studio code also assists in debugging and has embedded git integration.

Blender (Version 2.78)
Blender will be used for the creation of all 3D assets required for the prototype.

FL Studio
The Digital Audio Workstation software music production environment FL Studio will used for all music and sound creation.

8.2. Network Requirements

As this game is single player, has no multiplayer mode and requires one computer, no internet connection should be required for the game to be player.

8.3. Class Diagrams

CAMERAS

FocusCamera

- _originalFieldOfView: float
+ speed: float
+ rotateSpeed: float
+ offset: float
+ maxZoom: float
- _zoomObject: Transform
- _startZoom: bool
- _freeLookCamera: GameObject
- _originalTransform: Transform
- m_OnZoomFinished: UnityEvent

- Start() : void
- LateUpdate() : void
+ FocusOnObject() : void
+ ReturnToMainView() : void
+ ChangeFinishedZoomEvent() : void

FreeLookCam

- m_MoveSpeed: float = 1f
- m_TurnSpeed: float = 1.5f
- m_TurnSmoothing: float = 0.0f
- m_TiltMin: float = 75f
- m_LockCursor: bool = 45f
- m_VerticalAutoReturn: bool = false
- m_VerticalAutoReturn: bool = false
- m_LookAngle: float
- m_TiltAngle: float
- k_LookDistance: float = 100f
- m_PivotEulers: int
- m_PivotTargetRot: Quaternion
- m_TransformTargetRot: Quaternion

Awake() : void
Update() : void
- OnDisable() : void
FollowTarget() : void
- HandleRotationMovement() : void

InteractableObject

- _clickedOnce: bool
- _doubleClicked: bool
- _doubleClickTimer: float
- _doubleClickDelay: int
- _startTimer: bool
+ zoomInLimit: int
+ zoomOutLimit: float
+ startingZoomScale: float
+ myCameraRot: Vector3
+ lockRotation: bool
+ returnZoom: bool
+ hintSprite: Sprite
- _currActiveObject: bool
- _interactableObjects: List<GameObject>
- _focusCamera: GameObject
+ m_OnZoomFinished: UnityEvent

- Awake() : void
- Start() : void
+ Update() : void
- OnMouseOver() : void
- OnMouseExit() : void
- ClearClicks() : void
- MakeThisCameraTarget() : void
+ GetZoomInLimit() : float
+ GetZoomOutLimit() : void
+ GetStartingZoomScale() : float
+ SetCurrActiveObj() : void
+ IsCurrActiveObj() : void
+ GetAllInteractableObjects() : List<GameObject>
+ MakeThisCameraTargetExternally() : void
+ ChangeOnZoomFinishedEvent() : void

ChangeAnalogGameEvent

+ m_ChangeAnalogueGameDoneEvent: UnityEvent

- Start() : void
- Update() : void
+ NewMethChangeAnalogEndEventod() : void

ChangeAxeEvent

+ m_ChangeEvent: UnityEvent

- Start() : void
- Update() : void

ChangeHandcuffEvent

+ m_ChangeEvent: UnityEvent

- Start() : void
- Update() : void
+ ChangeHandcuffOnZoomFinishedEvent() : void

ChangeHandcuffQTEvent

+ numberOfCombos: int
+ m_OnSuccessfulCombos: UnityEvent
- _currNumberOfCombos: int

- Start() : void
+ CheckHandcuffSuccess() : void

ChoicesCanvas

- _showing: bool

- Start() : void
+ Hide() : void
+ NewMethod() : void
+ IsShowing() : bool
+ GetChoiceEventsByName() : OptionEventSetter
+ SetChoicesByName() : void

InteractableObjectEventChanger

+ m_ChangeEvent: UnityEvent

- Start() : void
- Update() : void

OptionEventSetter

+ choicesName: string
+ choiceXText: string
+ newField: UnityEvent
+ choiceOText: string
+ m_OptionO: UnityEvent
+ choiceSquareText: string
+ m_OptionSquare: UnityEvent

+ SetChoices() : void
+ GetName() : string

Option

+ myKeyCode: KeyCode
- m_OnOptionChosen: UnityEvent
+ m_OnOptionChosenDefault: UnityEvent

- Start() : void
- Update() : void
+ ChangeEvent() : void

XOption

- m_OnOptionChosen: UnityEvent

- Start() : void

11

QUICK TIME EVENTS

QTEEventChanger

+ m_ChangeSuccessEvent: UnityEvent

- Start() : void
+ ChangeSuccessEvent() : void

QuickTimeManager

+ m_OnSuccess: UnityEvent
+ m_OnFailure: UnityEvent
- _QTEInProgress: bool
- _callQuickTimeEvent: bool

- Start() : void
- Update() : void
+ CallFailure() : void
+ CallSuccess() : void
+ ChangeSuccessEvent() : void
+ ChangeFailureEvent() : void
+ CallQuickTimeEvent() : void
+ Hide() : void
+ Show() : void

QTEIcon

- iconSprites: List<Sprite>
- _chosenSprite: Sprite

- Start() : void
+ RandomiseIcon() : void
+ GetCurrIcon() : string

Timer

- newField: float
- _timer: float
- _startTimer: bool

- Update() : void
+ StartTimer() : void
+ ResetTimer() : void

SUBTITLES

Subtitles

- _currLines: List<string>
- _index: int

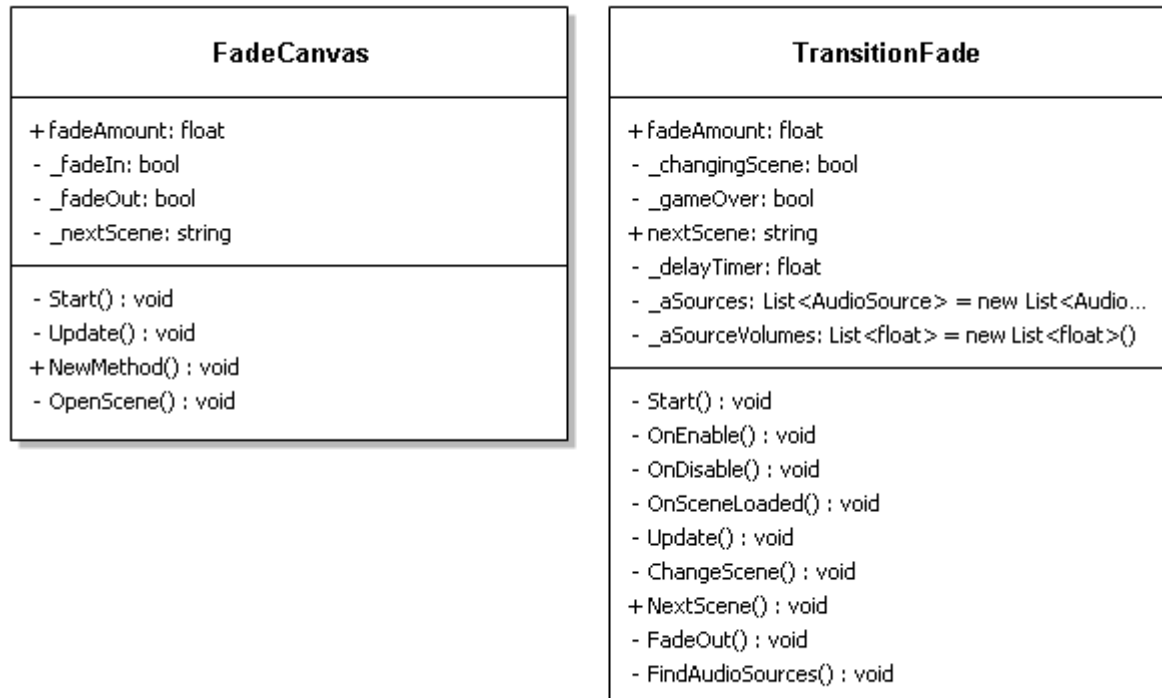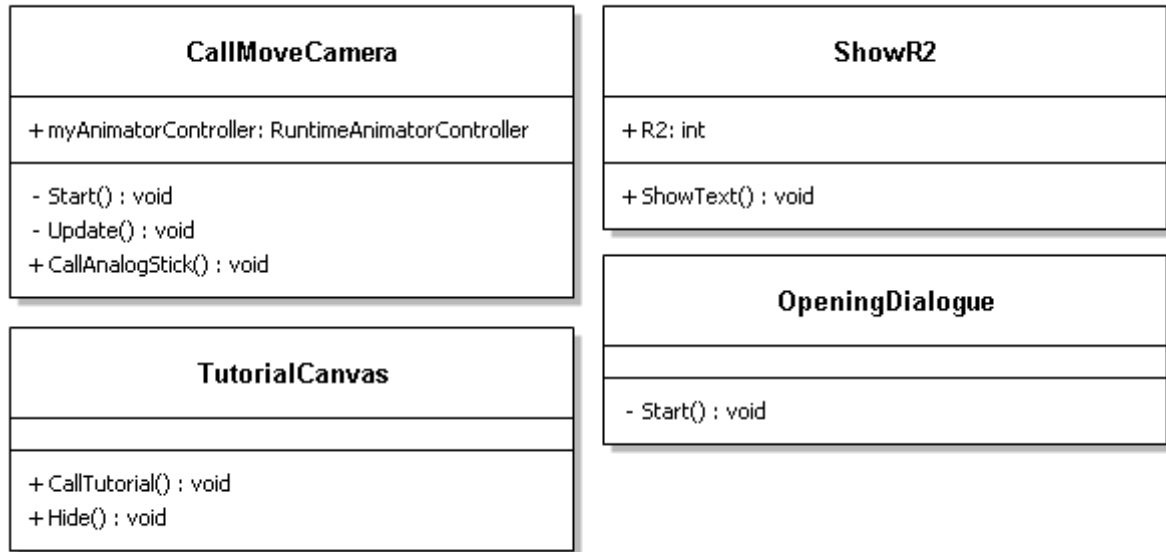- Start() : void
- Update() : void
+ NextLine() : void
+ SetCurrConversation() : void
+ Empty() : void
+ PlayConversation() : void
+ GetConversationByName() : Conversation

Conversation

+ conversationName: string
+ dialogueClip: AudioClip
+ animator: RuntimeAnimatorController
+ lines: List<string>

+ GetLines() : List<string>
+ GetName() : string
+ GetClip() : AudioClip
+ GetAnimator() : RuntimeAnimatorController

ToBeContinued

- Update() : void

DialogueSoundSource

+ PlayDialogue() : void

TRANSITIONS

FadeCanvas

+ fadeAmount: float
- _fadeIn: bool
- _fadeOut: bool
- _nextScene: string

- Start() : void
- Update() : void
+ NewMethod() : void
- OpenScene() : void

TransitionFade

+ fadeAmount: float
- _changingScene: bool
- _gameOver: bool
+ nextScene: string
- _delayTimer: float
- _aSources: List<AudioSource> = new List<Audio...
- _aSourceVolumes: List<float> = new List<float>()

- Start() : void
- OnEnable() : void
- OnDisable() : void
- OnSceneLoaded() : void
- Update() : void
- ChangeScene() : void
+ NextScene() : void
- FadeOut() : void
- FindAudioSources() : void

TUTORIAL TIPS

CallMoveCamera

+ myAnimatorController: RuntimeAnimatorController

- Start() : void
- Update() : void
+ CallAnalogStick() : void

ShowR2

+ R2: int

+ ShowText() : void

TutorialCanvas

+ CallTutorial() : void
+ Hide() : void

OpeningDialogue

- Start() : void

BACKGROUND

TreeMaker

+ tree: GameObject
+ gap: int
+ treeNum: int

- Start() : void

Road

- Update() : void
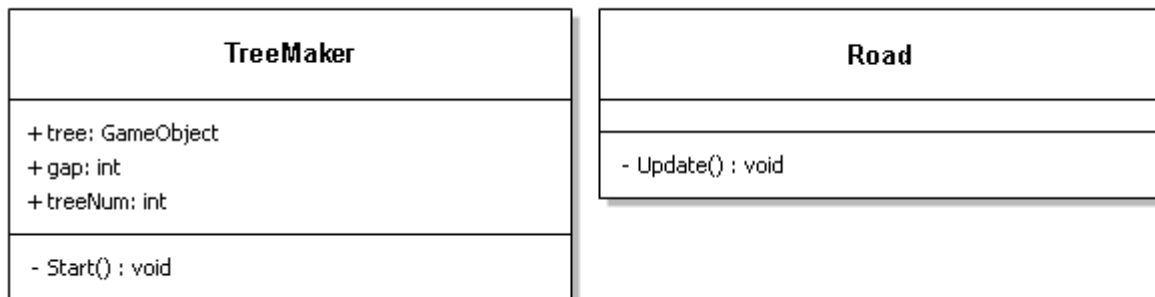
14

9. REFERENCES

1. Herrera, L. (2019). Retro Computer Aesthetic HD Wallpaper. [image] Available at: http://wallpaperswide.com/retro_computer_aesthetic-wallpapers.html [Accessed 28 Jul. 2019].

2. Nyamyam 2014, Tengami, video game, Japan

3. Oppland, M. (2016). Mihaly Csikszentmihalyi: All About Flow & Positive Psychology. [online] Positivepsychologyprogram.com. Available at: https://positivepsychologyprogram.com/mihaly-csikszentmihalyi-father-of-flow/ [Accessed 26 July 2019].

4. Rohlandt, E. (2019). PS4 Controller Button Layout. [image] Available at: https://dribbble.com/shots/3467476-PS4-Controller-Button-Layout [Accessed 28 Jul. 2019].

5. Sony Interactive Entertainment 2018, *Detroit: Become Human*, video game, USA

6. Sony Computer Entertainment 2015, *Until Dawn*, video game, USA

7. Toxel (2016). Paper Car. [image] Available at: http://www.toxel.com/inspiration/2016/09/07/paper-car/ [Accessed 28 Jul. 2019].