Interactive Audio Technical Notes:

Library sound effects:

I have used a variety of sound sources to create the final audio assets. My main sources were freesound.org, commercially released sound effect libraries and my own sound library. All sounds were manipulated either in frequency, pitch or volume and often combined together to form a new soundscape. This process mainly took place in Pro Tools and Reaper and I then imported the final audio tracks into Wwise.

Ambient:

The ambient sounds were mainly modified in volume and frequency shape.
Forest: Most sounds for the day cycle are from the “Originals” folder of the Wwise project. The only addition is the bird chirping track, which is from the “General HD” collection of “Sound Ideas” and the cricket sounds I have recorded myself. For the night cycle I have used the same approach and used one sound from the “General HD” library and the wolf howling is from a free sound effect by George Vlad´s “Mindful Audio”. I found the music advertised as royal free through Youtube and the composers website. The day music is called “Riddle of the Forest” by Martynas Lau and the night music is called “The Forbidden Forest” by Alexander Hoff.
Sea and Shore: I have found the sea sounds on “freesound.org” and in a collection called “Soniss GDC 2018 Free Game Audio Bundle”. The shore sounds come from my own library. I found the music advertised as royal free through youtube and the composers website. The day music is called “Pirate Crew” by Ross Budgen and the night music is called “Zen Hand Drum” by “DL-Sounds”.
Town and Town_Market: Most sounds for the day cycle are from the sound library “Medieval Life” by the company “Boom Library”. An exception is the windmill sound, which is from the “sfx kit” by “Sound Ideas”. Most sounds for the night cycle are sourced from the “Originals” folder within the Wwise folder. In addition to that I used the same wolf howling track from “Mindful Audio” as in the forest and the wind track is from “freesound.org”. I found the music advertised as royal free through Youtube and the composers website. The day music is called “Village Consort” by “Always Music” and the night music is called “Medieval Courtyard” by “TeknoAXE”.

Character:

The footstep and jump sounds are sourced from the “Originals” folder within Wwise, an asset folder given to us by John and from the “General HD” library from “Sound Ideas”. Additionally I have recorded some of the rooftop footsteps sounds myself.

Emitter:

Ambience: The torch sound effect was provided by John.
SFX:
Dragon Tower: The ambience sounds and one of the dragon vocal sounds were sourced from “Soniss GDC 2017 Free Game Audio Bundle”, the other

dragon vocal sound is from “the SFX kit” by “Sound Ideas”. Both were lowered in pitch to fit the visual appearance. I have found the music on “freesound.org” titled as “Epic Orchestra” by Joshua Empyre.
Tiger Jungle: Most of the ambient sounds are recordings I made in Thailand in conjunction with a clip from the “Sound Effects Bible”. The tiger sounds are from the “Amadeus Sound Effects Library” by “Sound Ideas”, the “General HD” library by “Sound Ideas” and “Bengal Tiger” by “Pro Sound Effects”.
Market Emitter: As stated above, all sounds for the market ambience are from the sound library “Medieval Life” by the “Boom Library”.
Pick-Up: The cymbal hit is from the “Music” folder in the “Originals” folder within the Wwise project folder.
Ship_Casual: The sounds are from the “Sound Effects Bible”.
Ship_Combat: The sounds are sourced from “freesound.org”, the “General HD” library and “Detroit Chop Chop”. I found the music advertised as royal free through Youtube and the composers website and it is called “Seas at War” by Jay Man.
Tavern: The ambient sounds are from “freesound.org”, the “General HD” library and “Sound Pack Tree”. I found the music advertised as royal free through Youtube and the composers website and it is called “The Red Fox Tavern” by Curran Son.
Zombie Temple: The wind sounds are both from my own recordings and from the “SFX kit” by “Sound Ideas”. The bell is from the “General HD” library and zombie sounds are both my own voice processed and from the library “Magic Elements” by “Articulated Sound Effects. I found the music advertised as royal free through Youtube and the composers website and it is called “Dark Tension Rising” by Mattia Cupelli.
Whale: The whale sounds are from the “General HD” library and the paddling sounds are both my own recordings and from the library “The Sound of Water” by “Tonsturm”.
Music: The Christmas music was advertised as royal free and is called “Christmas Instrumentals” by “heroboard”.
Vocal: The voice over is my own voice lowered in pitch.
Music_Horizontal: The music stems are sourced from the “Music” folder within the “Originals” folder of the Wwise project.

Environment Sound Design:

Each ambient location has different set of ambient sounds and background music for either day or night. I used the following ambient locations: Ambient_Forest, Ambient_Sea, Ambient_Shore, Ambient_Town, Ambient_Town_Market. In order to do that I have attached all ambient location containers in Wwise to the game parameter “time of day” and created blend tracks in Wwises to create smooth transitions. Both the day and night sounds consist of background loops and randomized containers played together. I have sourced at least 3 different elements for a particular ambience and put them in a random container. I have added a silence clip to generate naturalistic pauses and create a believable environment. Moreover, the ambience containers have randomized volume levels and sometimes randomized pitch values to make it even more non-repetitive. The background music always represents the mood of the location, for example the forest has eerie music while the sea has calm music.

Next to creating a suitable event and SoundBank in Wwise I have created states for each ambient location, which are routed in the Master Bus Hierarchy to create stop and play states. In Unity I have created box colliders for each location and attached both AkAmbient and AkState scripts which a triggered when the collider is entered by the player. Next to attaching the SoundBanks in Unity each ambience locations has three states: one “play state” and two “stop states”, which start and stop the correct environment sounds for each collider. I did this for both ambience sounds and ambience background music. To implement the sounds I have created the appropriate events and SoundBanks in Wwise. In Unity I loaded the SoundBanks and attached the “Time of Day” parameter to the “TimeController” script (line 38).

Event Sound Design:

Character:

In Wwise I have set up one switch container for every surface type in game level, which are “Dirt_Rock”, “Dirt_Sand”, “Water”, “Wood_Plank”, “Wood_Roof”, “Wood_Straw”. Each of those containers consist of two random containers, one for running and one for walking, both include 4 different footstep sounds to make it non-repetitive. The surface containers are attached to the “Step_Type” switch I have created in the “Game Syncs” tab.
The Jump vocalizations are also in a random container, which makes it more naturalistic. Additionally I have checked the box “Use game-defined auxiliary sends” to make use of reverb environment in Unity. Last but not least I have created the necessary events and SoundBanks.
In Unity I have attached my different surface types to the “Footsteps collider” script and attached it to the FPSController object. Additionally I had to add AKSwitch scripts for each surface type and load the appropriate SoundBanks. To implement the jump vocalizations I had to load the SoundBank and attach the Wwise event to the “FPSController” script (line 149-156). Moreover, I had to edit the line 173 so that there is no footstep sound playing while being mid air after the player jumped. To have an appropriate landing sound I modified the line 93-94 so that the footstep sounds are slightly delayed, which makes it more naturalistic. Last but not least I had to attach the “Material Switch Controller” script to the appropriate GameObjects to trigger the right surface sounds at the right moment.

Emitter:

Ambience: For the torch sound I have created a suitable attenuation within Wwise and set the 3D Spatialization to “Position+Orientation” to create a realistic environmental representation. Last but not least I generated the necessary events and SoundBanks in Wwise.
In Unity I loaded the SoundBank and attached an “AkAmbient” script to each torch GameObject, which is triggered on Start.

SFX:

Dragon_Tower: Consists of a background ambience loop, a background music loop and a random container for the dragon vocalizations. Except the music the sounds make use of “game-defined auxiliary sends” and the dragon container is placed in the 3D Spatialization. All three components of the Dragon_Tower asset are attached to their RTPC parameters to. I have added both a Voice volume and Low-Pass effect to let the sounds become more

audible when the player gets closer to the object. Last but not least I have created the appropriate events and SoundBanks.
In Unity I loaded the SoundBanks and attached AKAmbient scripts and a sphere collider to both the tower and the dragon, which are triggered when the player enters the collider. For the reverb I have created a capsule collider around the tower and attached an AKEnvironment script, which triggers the Reverb Bus I set up in Wwise. Last but not least I created a “Distance to Checkpoint” script for the game parameters “Tower” and “Dragon” and attached it to the appropriate GameObjects and inserted it into the “FPSController” GameObject.
The same approach was used for the following SFX assets and their corresponding music assets: SFX_Jungle_Tiger, SFX_Market_elements, SFX_Ship, SFX_Ship_Combat, SFX_Tavern, SFX_Temple_Zombie and SFX_Whale.
There are some distinctions: For the “Jungle” ambience I had to set up a box collider in Unity to attach the scripts to, similar to the Ambience locations colliders. Except for the “Temple_Zombie” asset non of the above mentioned assets make use of Reverberation. For the “Temple” GameObject I created an capsule collider for the reverb zone and proceeded the same way as at the “Tower” event.
Pick Up locations: I have created a mini game within the game level with 8 different locations the player has to go and collect an object. In Wwise I created both a pick up sound effect asset and a voice over specific for each location.
In Unity I first loaded the appropriate SoundBanks. Then I created an UI Text GameObject, which displays how many locations are left. Next I created a cube GameObject for each location, which is the pick up object. I have added a mesh collider, which is set to trigger, and attached an AkAmbient script for the pick up sound, which is set to “TriggerEnter”. Additionally I added a “Objects to collect” script so that the game recognizes a collision. In the “EventSystem” Object I added the “CountObjects” scrip so that the UI text is updated in real time and displays how many locations are left.
Next to the pick up object is a GameObject, most of the time it is a knight, which triggers the voice over, The knight has a sphere collider and an AkAmbient script, which is set to “TriggerEnter”.
The final location has a different sound attached to the cube pick up object, which is the “christmas music”.

Interactive Music:

Vertical Music: I treated the vertical background music the same way as the ambience assets and mentioned it in the “Environment Sound Design” and “Event Sound Design” section.
Horizontal Music: I created a compelling music track when the player gets close to a pick up location. In order to do that I set up an individual “Music Switch Container”, a “Music Playlist Container” and “Music Segment” for each location. In addition I created an individual state for each music location and a game parameter called “close_to_pick_up” for the RTPC settings. I attached the states to their switch containers in the association editor and assigned the playlists, which are set to “sequence continuous”. Each “music segment”

consists of three music stems, which are triggered by the distance between the player and the pick up object. I assigned the RTPC values to the game parameter and created the appropriate events and SoundBanks.
In Unity I loaded the SoundBank and created box colliders for each horizontal music location. Moreover, I have attached an AKState script, which is set to "TriggerEnter" and chose the appropriate state. Next I created a "Distance To Checkpoint" script for each location, changed the parameter in line 31, and attached it to the right pick up GameObject and in the end added it to the "FPSController" GameObject.

Interactive Mix:

States: As mentioned before, I created "play" and "stop" states for my ambient assets, which are routed to their appropriate Bus in the "Master-Mixer-Hierachy" and have dedicated volume and transition volumes set up in the property editor.
Ducking: I have used ducking several times, for example when the player approaches the "Ship_Combat" event the "Ship_Casual" asset is ducked in the "Master-Mixer-Hierachy". Another example is the horizontal music, which ducks the vertical music whenever it plays. Moreover, the voice lines duck the ambient asset and music assets. Last but not least, the final music track ducks all other assets.
Reverb: As mentioned before, I created two reverb busses, one for the "Temple" location and one for the "Tower" location. In Unity the bus is then triggered by an AKEnvironment script.
Soundcaster and Mixing Desk: I created separate "Soundcaster Sessions" for each event and ambience location and mixed all the assets together. Afterwards I moved to the "Busses_Mix" in the "Mixing Desk" and created the final mix.

Annotations:

Horizontal Music: I wished the music would not play after the pick up object is destroyed, but I could not get it to work. At the moment I have timed the the horizontal music to go off when approaching the pick up object in walking pace.
Bug: My work starts properly after around 10 seconds after entering the game in Unity. Before that random sound appear and weird volume levels play.

Links:

Blog with demonstration video:
http://www.nilsgradlowsky.com/game-audio.html