Survive or Die Trying Audio

Survive or Die Trying is an open world survival game located on a remote island. The player must collect resources to craft tools that will aid them in their endeavors. While trying to keep from starving or dehydrating, the player must also be wary of the more aggressive wild life on the island. The goal is for the player to collect resources to make a raft to escape the island alive.

The game required sounds for player movement, such as walking on various surfaces, and player actions, such as eating and cutting down trees. Sounds for the animals, such as the bear attacking, and the ambience sounds, such as trees groaning in the wind and waves, also needed to be created. The main menu and pause menu also both required UI sounds. Overall, I believe I fulfilled all the requirements for the project. I created multiple sounds and events for various regions on the island to create an immersive ambience. I also made sounds for the player walking on the different surfaces on the island, as well as dynamic splashing sounds for when they are in different depths of water. The main menu and pause menu both have UI sounds that help the player navigate the menus better. Lastly, the player has action sounds that range from eating and drinking to stabbing an animal with a knife.

Most of the source files I used to create my sounds only required minor adjusting of pitch and volume with a few exceptions. The first sets of sound I created for the game were all the stepping sounds for the various surfaces. This required the fine adjusting of the pitch of each sound to make them sound like that all were created in the same environment. The forest and beach ambience sounds as well as the birds required me to filter out some pitch levels of different source sounds in order to create a realistic blend when they were played together. These files, along with a few others, also required fine-tuned splitting in order to create a seamless loop when played back. Lastly, all the sounds that had dialog or vocalizations in them required more work than most the other assets to make them sound like they all originated from the same person. This required me to pitch shift several of the audio samples as well as adjusting the pitch with filters.

In FMOD, my events are organized into four main categories which are ambience, dialog, sound effects, and UI. Within the ambience category, there are three smaller divisions two correspond to the map regions and one is for the birds. Within the category of the beach, the first event is seagull. I used a scatterer instrument for this so that the sounds can be spawned randomly around the player to make it more realistic. I also set the start offset to be completely random so that it is harder to notice the repetition. The other event in this category is the waves sound, which is a 3D sound event that is looped continually, which I then created instances of in unity and placed at various points off shore to make the waves sound realistic. In the bird category, there is only one event with a scatterer instrument in it. I did this so that the different bird sounds will be placed randomly around the map to make the player feel more immersed. The last ambience category was the forest which had one event with a scatterer instrument that player tree groaning sounds in the 3D space around the player when they were in the forest to make the forest feel more realistic.

The next main category that I organized my events into was dialog, which only had three dialog events in it. The first dialog event was to inform the player that they were hot, but the temperature functionality was not implemented in the game yet so it was not used. The event contained a single instrument with no randomization. The second dialog event was informing the player they were hungry and it had a single instrument that plays the dialog when the player falls below a certain hunger. I was originally going to use a parameter within FMOD to control when this event would be called, but was unable to figure out how to get the parameter to work with the hunger system we already had implemented in the game. This is the same issue that I had with the thirst dialog event and to fix this, I simply called the events from a script within the game.

The third, and largest, category that I organized my events into was the sound effects group. The first set of events in the category belong to the animal group, more specifically the bear. The first bear audio event was simply the bear growling which I crated using a single instrument with some pitch and volume randomization for realism. This is the same procedure that I followed for the creating the bear roar event, which is the second bear event. Within the sound effect group, there is also an environment sound group that has the tree falling sound event in it. This sound has a single instrument with a small amount of pitch and volume randomization, for realism, that is attached to a tree and plays in the 3D space when the tree is cut down. The last group within the sound effect category is the player group, which is broken down into damage, interact, movement, and tools categories. In the damage category, the death event, physical damage event, and vomit event are simply made up of unmodulated single instruments. I chose not to modulate these events because I wanted the vocalization pitch to be unchanged so that I sounds like the vocalizations are coming from just one person. The drowning event is a single instrument that is looped when so it plays continually when the player is in the drowning state. In the interact category, the collect plant, eat, and drink events all contain single unmodulated instruments. I chose to not edit these events because they are used to give the player information about an action they have done, so it works better when it is a repetitive sound. The next group within the player is the movement group which has the fall and walking group within it, as well as the jump event. The jump event contains a single instrument with slightly volume modulation to make the sound less repetitive. In the fall group, the fall without damage event has an event instrument that plays that step event at a louder volume to make it sound like the player is landing harder. The fall with damage event has the step event instrument in it as well, but it also has a single instrument that plays the grunt sound as the same time as the loud step sound. I did it this way so that the material that the player is landing on can be set easier through the step event. The last event group in the player movement group is the walking group. The step event in the walking group has three multi instruments, which contain several step sounds, and an event instrument that references the water depth event. The four instruments correspond to the surface types that the player can walk on and which instrument is playing is controlled by a parameter that can be set while playing. All the multi instruments also have a small amount of pitch and volume modulation to add to the realism. The swimming event is a single looped instrument with a fully randomized start offset to lower repetition. The water depth event contains two multi instruments, one which has puddle step sounds and the other which has wading sounds. The playback of the instruments is controlled by a parameter called water depth that is set within the game while playing. I blended the transition of the two events when the water depth is between .2 and .8 to make the sound more realistic as well as making the transition smoother sounding. Finally, the last group within the player group is the tools group which contains two events. The first event is the axe swing event, which contains three single instruments that have pitch and volume modulation for realism. The playback is controlled by a parameter that is set in the game based of what the player is hitting when they swing the axe. The second event is the stab sound which has a single instrument with some pitch and volume randomization for realism.

Lastly, the final main category that I organized my events into was the UI category. The three events in this group, button click back, button click, and crafting, are all simply made up of single unmodulated instruments.

The mixing of my project required me to adjust the levels of some of the events, such as lowering the walking and bear roaring sounds, as well as adjusting the audio asset files of several of the ambience sounds to make the mix sound more realistic. I also created several snapshots that change the ambience volume levels corresponding to different regions or times of day.

For the mastering of my game, I compared the LUFS of my game against two games of similar style and genre. The first game was Raft, which had -35.9 LUFS integrated, and the second game was Green Hell, which had -22.3 LUFS integrated. When I measured my game, my integrated LUFS was at -23.9 which is within the range I feel it should be based off the other games. Since the gameplay of this game is much more similar to Green Hell than Raft, the closeness of their integrated LUFS is great.

Sources

stone-step.wav: EricsSoundschmiede <https://freesound.org/people/EricsSoundschmiede/sounds/476697/>

dirt-step.wav: [FocusBay https://freesound.org/people/FocusBay/sounds/210865/](FocusBay%20https://freesound.org/people/FocusBay/sounds/210865/)

sand-step.wav: byjoshberry <https://freesound.org/people/byjoshberry/sounds/431416/>

water-step.wav: EricsSoundschmiede <https://freesound.org/people/EricsSoundschmiede/sounds/454623/>

wading.wav: Motion\_S <https://freesound.org/people/Motion_S/sounds/221764/>

swimming.wav: 14GPanskaValaChristoffer <https://freesound.org/people/14GPanskaValaChristoffer/sounds/420167/>

swing.flac: qubodup <https://freesound.org/people/qubodup/sounds/60013/>

tree-chop.wav: hazure <https://freesound.org/people/hazure/sounds/23700/>

axe-miss.wav: Kane53126 <https://freesound.org/people/Kane53126/sounds/257928/>

bear-growl.mp3: Nivatius <https://freesound.org/people/Nivatius/sounds/519518/>

bear-roar.wav: aesqe <https://freesound.org/people/aesqe/sounds/49127/>

forest-background-1.wav: spoonbender <https://freesound.org/people/spoonbender/sounds/244942/>

forest-background-2.wav: kangaroovindaloo <https://freesound.org/people/kangaroovindaloo/sounds/147638/>

person-hit.mp3: RICHERlandTV <https://freesound.org/people/RICHERlandTV/sounds/232358/>

vomit.flac: D.jones <https://freesound.org/people/D.jones/sounds/528790/>

grunt.aiff: snaginneb <https://freesound.org/people/snaginneb/sounds/129346/>

wet-ground-hit.wav: kyles <https://freesound.org/people/kyles/sounds/450830/>

metal-ting.wav: Griphen <https://freesound.org/people/Griphen/sounds/366351/>

hammer.wav: InspectorJ <https://freesound.org/people/InspectorJ/sounds/406048/>

bite.wav: InspectorJ <https://freesound.org/people/InspectorJ/sounds/331395/>

slurp.wav: SeanPorio <https://freesound.org/people/SeanPorio/sounds/519268/>

death-cry.wav: theuncertainman <https://freesound.org/people/theuncertainman/sounds/417539/>

crunch.wav: InspectorJ <https://freesound.org/people/InspectorJ/sounds/333818/>

drowning.wav: sarson <https://freesound.org/people/sarson/sounds/144201/>

waves.wav: szegvari <https://freesound.org/people/szegvari/sounds/530699/>

tree-falling.wav: Kinoton <https://freesound.org/people/Kinoton/sounds/494071/>

tear.wav: SophieMezaM <https://freesound.org/people/SophieMezaM/sounds/446096/>

hup.aiff: enmiller2 <https://freesound.org/people/enmiller2/sounds/171659/>

birds1.mp3: jmiddlesworth <https://freesound.org/people/jmiddlesworth/sounds/364663/>

birds2.wav: InspectorJ <https://freesound.org/people/InspectorJ/sounds/339326/>

seagull.mp3: squashy555 <https://freesound.org/people/squashy555/sounds/353416/>

stab.mp3: Podcapocalipsis <https://freesound.org/people/Podcapocalipsis/sounds/521030/>