UVU Studios

Design Document for:

Star Catcher

“Gotta Catch ‘em All”

All work Copyright ©2016 by Utah Valley University

Written by Cassidy Barlow

Version # 1.20

Saturday, October 1, 2016

**Table of Contents**

[Star Catcher 1](#_Toc463128245)

[**Table of Contents** 2](#_Toc463128246)

[**Design History** 4](#_Toc463128247)

[**Version 1.10** 4](#_Toc463128248)

[**Version 1.20** 4](#_Toc463128249)

[**Game Overview** 4](#_Toc463128250)

[**Story**  **(if there is one)** 5](#_Toc463128251)

[**Feature Set** 5](#_Toc463128252)

[**Core Game Play** 5](#_Toc463128253)

[**Game Flow** 5](#_Toc463128254)

[**Characters** 5](#_Toc463128255)

[**Game Play Elements** 6](#_Toc463128256)

[**Game Physics and Statistics** 6](#_Toc463128257)

[**Artificial Intelligence** 6](#_Toc463128258)

[**~~Multiplayer~~** 6](#_Toc463128259)

[**User Interface** 7](#_Toc463128260)

[**Flowchart** 7](#_Toc463128261)

[**Functional Requirements** 7](#_Toc463128262)

[**Mock-ups** 8](#_Toc463128263)

[**~~GUI Objects~~** 9](#_Toc463128264)

[**Art and Video** 10](#_Toc463128265)

[**Overall Goals** 10](#_Toc463128266)

[**~~Level Design Seeds~~** 10](#_Toc463128267)

[**~~Level Diagram~~** 10](#_Toc463128268)

[**~~Cinematics~~** 15](#_Toc463128269)

[**~~Video~~** 16](#_Toc463128270)

[**Sound and Music** 16](#_Toc463128271)

[**Overall Goals** 16](#_Toc463128272)

[**Sound Effects** 16](#_Toc463128273)

**Design History**

This is a description of each version of the game as it updates, in order to keep track of changes.

**Version 1.10**

Version 1.10 includes the creation of the first game prototype.

1. I have modeled each land module and added them to the unity scene, with only land module 6 being unable to work properly.
2. I created a capsule to represent the player and incorporated scripts made in Anthony Romwell’s class to the player and camera. The module recycler isn’t working and Anthony said to omit it for the time being.
3. Created the Design document and updated it on 9/22/16

**Version 1.20**

Version 1.20 includes a new project folder for the prototype.

1. I created a new unity project folder and imported the necessary assets to reduce clutter and make navigation easier.
2. Using the same scene to recycle cubes in Anthony’s class, I have added land modules and rearranged my distance and position variables to accommodate for their proportions.
3. I added colliders to the camera to restrict movement outside of the camera view and to the creviced land modules. I then created a Kill Player script add attached them to the cubes/colliders parented under the crevices and Left Camera Collider.
4. In the Kill Player Script, I added a function that resets the scene when the player is killed (this seems to affect the materials of the modules for some reason). This acts as a respawn that resets the placement of both the player and land modules. In order for this to work I had to place my previously static variables of “distance” and “nextSectionPosition” inside the recycleLevelObject script so that the land modules next recycle position would reset with the scene.

**Game Overview**

The main premise of this game is to catch as many stars as possible while attempting to survive in an endless runner setting. This game is single player, in a desert setting, where the player controls the rabbit to collect the stars and avoid obstacles. The player will also have to watch out for enemies such as the wolf and hawk in order to keep stars and maintain health. This game is unique because it requires the player to stay attentive and engaged in order to make it as far as possible.

**Story (if there is one)**

The original story followed the Navajo Black God myth, and was going to have the rabbit collecting stray stars that are falling from the sky after Coyote tossed them from Black God’s buckskin. However, as we do not have permission to use this story, we are simply having a rabbit collecting stars and trying to avoid a wolf and other enemies.

**Feature Set**

**Core Game Play**

* Running
* Collecting stars
* Avoiding enemies/obstacles to increase score and health

**Game Flow**

The player will first open the game by selecting the game shortcut. A main menu will come up as soon as the game is loaded. From the main menu, the player can then select either “Start Game” to start the game or “Options” to toggle with the options settings. In the options screen, the player can select the back arrow icon to return to the main menu and start the game. Once the game screen has loaded, the player will see an icon saying “Select Any Key to Begin”, and once selecting with the left mouse button, commands the character and camera to begin movement. The player must then use commands to move the player left, right, and to jump and double jump within the screen while the camera moves forwards in a progressive but steady pace. On the game screen the player can also use the mouse to select the pause icon, which will bring up a menu that will allow access to either the options screen, or allow them to return to the main menu and reset the game.

**Characters**

* Rabbit
  + The rabbit is the character that will be controlled by the player. This character will have all of the basic movements, including moving left/right, jump, double jump, and a speed boost/slide.
* Wolf
  + The wolf is the main enemy that the player will have to avoid. He will appear at the edge of the screen, and will either rush at the player, or pounce.
* Hawk
  + The Hawk is an enemy that will appear from the top of the screen and swoop down at the character.

**Game Play Elements**

* Health Hearts
  + Health hearts are elements that can either replenish or boost the player characters’ health when they collide with the character’s collider
* Stars
  + Stars are elements that fall randomly from the top of the screen and are collected by the player character upon collision, and accumulate as the score total.

**Game Physics and Statistics**

* Star Collision
  + Characterized by sound effects and a particle system, which will be gold in colour.
* Stars Falling
  + The stars will fall randomly from the top of the screen at different speeds generated randomly by the code.
* Collision with Wolf and Hawk
  + When the player character collides with the enemies, the player will be moved backwards a few units and bounce before regaining forward movement.

**Artificial Intelligence**

The AI must be able to generate stars randomly in front of the player. The Camera will act as the trigger, and when it collides with areas that appear on screen it will randomly generate stars at the right side of the top of the screen. there must also be a limit to the number of stars generated for a certain amount of time.

Enemies will be triggered by the score counter, and when the player reaches an X amount of stars an enemy will be triggered to appear on screen and attack the player. As another idea, enemies could also be triggered by the camera movement, with the triggers being generated randomly, but not constantly, so that the enemies appear in random intervals.

**~~Multiplayer~~**

~~Indicate the methods of multi-player play (i.e. head-to-head, cooperative vs. AI, teams, every man for himself, hot seat) and how many players it will support on the various networking methods. Describe how multi-player differs from solo-play in game flow, characters/units, game play elements and AI.~~

**User Interface**

**Flowchart**

**Functional Requirements**

* Main Menu
  + User Actions – buttons, clicks, and hotspots for options and gameplay screens, with a fade out animation to that screen.
* Gameplay Screen
  + User Actions – buttons, clicks, and hotspots for the pause menu, fade in/ pop up animation for the menu coming into the screen
* Pause Menu
  + User Actions – buttons, clicks, and hotspots for options, main menu, and gameplay screens, with a fade out animation to that screen.
* Options Screen
  + User Actions – buttons, clicks, and hotspots for main menu and pause menu screens, with a fade out animation to that screen.

**Mock-ups**

* Main Menu
* Gameplay Screen
* Pause Menu
* Options Screen
* Death Screen

**~~GUI Objects~~**

~~These are the basic building blocks used to create all the screens, windows and menus. This should not include the items seen in the main view portal, as these are covered in the art list in the next section. The GUI objects are primarily listed here for the programmers to know what pieces they’ll need to code and have for putting together the screens. You should explain in detail how each is interacted with and how they behave. It may seem a bit obvious and not worth documenting, but it really helps when drafting together the technical spec and schedule to know exactly everything the game will need.~~

~~For some games, this can be a very quick list to put together – buttons, icons, pointers, sliders, HUD displays etc. But it’s much more complicated in games where the interface is at all different. However, keep in mind that the methods of interaction are not all that different. A button is still a button, even if it’s clicking on a gorgon’s head instead of a grey rectangle.~~

**Art and Video**

**Overall Goals**

The art style for this game need a realistic tone to match that of the modelled land modules the we created in Maya. They don’t have to be hyper realistic, but they need to tie in together nicely enough as not to distract the player from the gameplay.

**~~GUI~~**

~~Screens, windows, pointers, markers, icons, buttons, menus, shell etc.~~

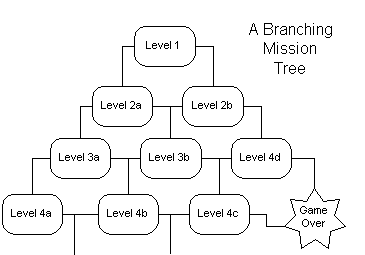
**~~Level Creation~~**

**~~Level Design Seeds~~**

~~These are the seeds for the detailed paper designs to follow. Detailed paper designs at this point are less legitimate and unlikely to survive intact. Designs created after the designers have had time to experiment with the tools and develop the first playable level are much more likely to succeed. It’s best to just plant the seeds for each level with a description of the goals and game play and where it ties into the story (if applicable). A thumbnail sketch is optional, but very helpful if the designer already has a clear idea of what he or she wants. Be sure to list any specific requirements for the level, such as terrain, objectives, the revelation of new assets, and target difficulty level.~~

**~~Level Diagram~~**

~~Whether this is a linear campaign, a branching mission tree, or a world-hopping free-for-all, this diagram should be the backbone upon which all the levels are built. A diagram isn’t necessary if the structure is so simple that a list would suffice. The following is an example of a typical success/fail branching mission tree. Of course this will vary greatly for each game. The important thing is that it just presents a road map for the level designers and for the readers.~~



**Terrain**

The environment used for this game will be desert based, using plateaus, chasms, and slopes.

Module 1:

* This Module will be flat with cacti and rocks acting as the only obstructions to the player's path.

Module 2:

* This module will slope to the left, with cacti and rocks as obstructions, making it so that the character seems to be running uphill.

Module 3:

* This module with slope to the right, with cacti and rocks as obstructions, making it seem like the character is running downhill.

Module 4:

* This module will have a chasm on the left side that the character must jump over (possibly double-jump over?)

Module 5

* This module will have a chasm on the right that the player must jump over.

Module 6

* This module with have a plateau in the middle that the player will have to scale and jump down from.

Module 7

* This module with have a plateau on the left that the player will have to scale and jump down from.

Module 8

* This module with have a plateau on the right that the player will have to scale and jump down from.

**~~Game Play Elements~~**

~~Player and enemy animations (sprites or models), game play structures and interactive objects, weapons, power-ups, etc. Don’t forget damage states.~~

**2D and 3D Assets**

|  |  |  |  |
| --- | --- | --- | --- |
| **3D Assets** |  | **2D Assets** |  |
| Asset Name |  | Asset Name |  |
| Environment Module 1 |  | Rabbit (player) |  |
| Environment Module 2 |  | Wolf (enemy) |  |
| Environment Module 3 |  | Stars |  |
| Environment Module 4 |  | Sky Background |  |
| Environment Module 5 |  | Hawk (enemy) |  |
| Rock Objects (?) |  | Pause Button |  |
| Cacti Objects (?) |  | Start Button |  |
|  |  | Options Button |  |
|  |  | Main Menu |  |
|  |  |  |  |
|  |  |  |  |

**~~Asset Revelation Schedule~~**

~~This should be a table or spreadsheet of what level the game’s assets are to be revealed to the player for the first time. There should be a row for each level and a column for each general type of asset. Assets include power-ups, weapons, enemy types, tricks, traps, objective types, challenges, buildings and all the other game play elements. The asset revelation schedule ensures that assets, the things that keep the players looking forward to the next level, are properly spaced and not over or under used.~~

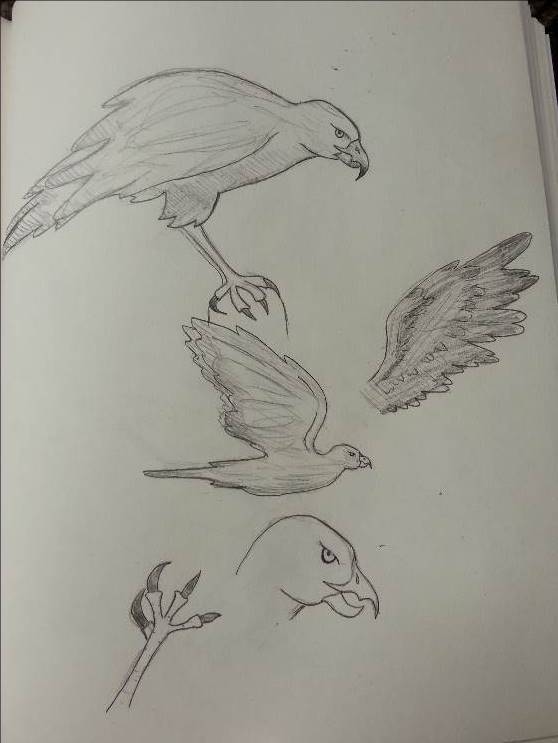
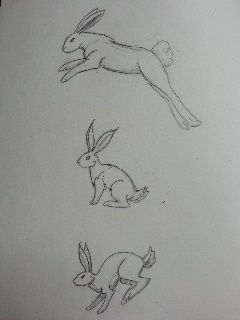
~~If it’s important to the game that certain assets stop being used, then the schedule might be better drawn as a Gantt chart with lines indicating the availability of assets. This gives the level designers a guide to what assets they have to work with so they don’t ruin their level or anyone else’s.~~

**Visual Effects**

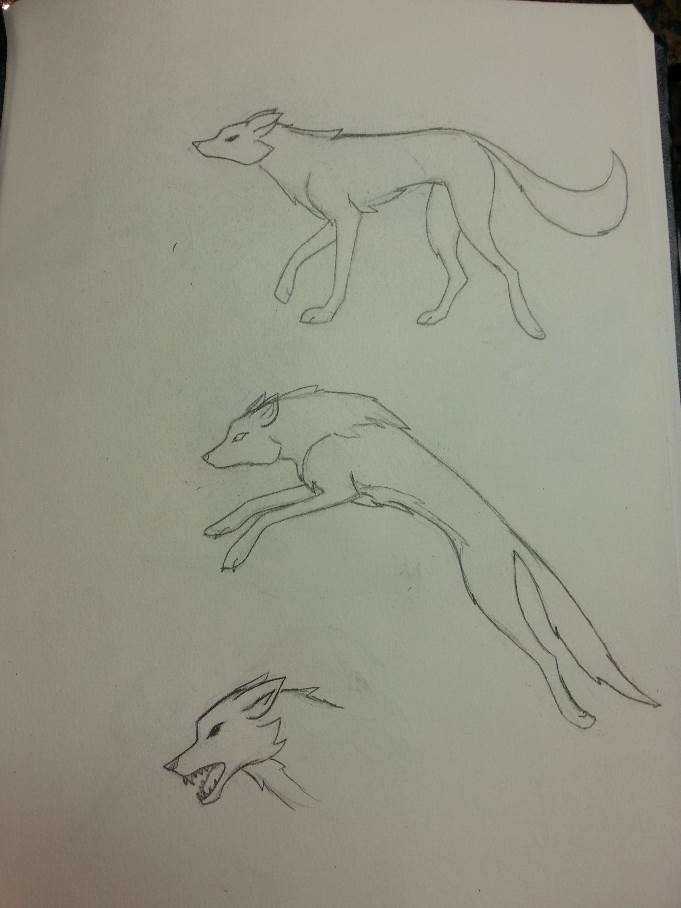
**Visual Effects**

* Stars being collected – gold particle system.
* Stars colliding with the ground – red particle system
* Stars being knocked away on collision with enemy – red particle system

**2D Art & Animation**

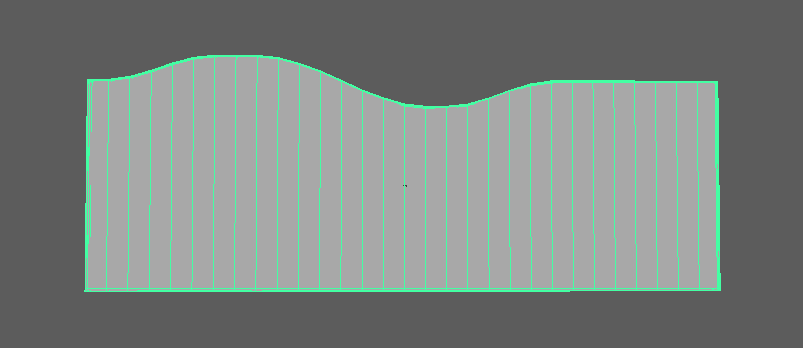


These are concepts of the characters that will be in this game. I was going for more of a realistic approach to their design, since the background and land modules will be more realistic.

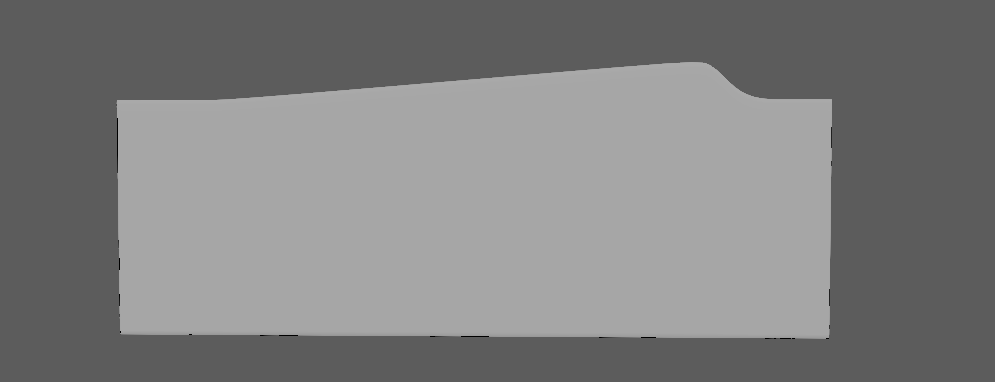


**3D Art and Animation (Land Modules found in project folder)**

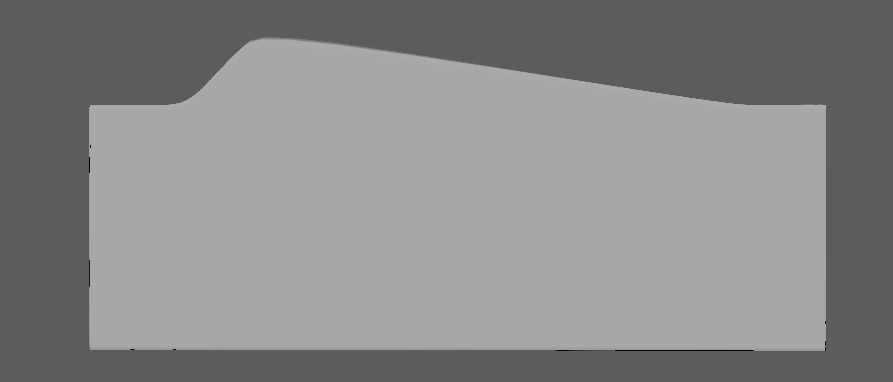
* Land Module 1



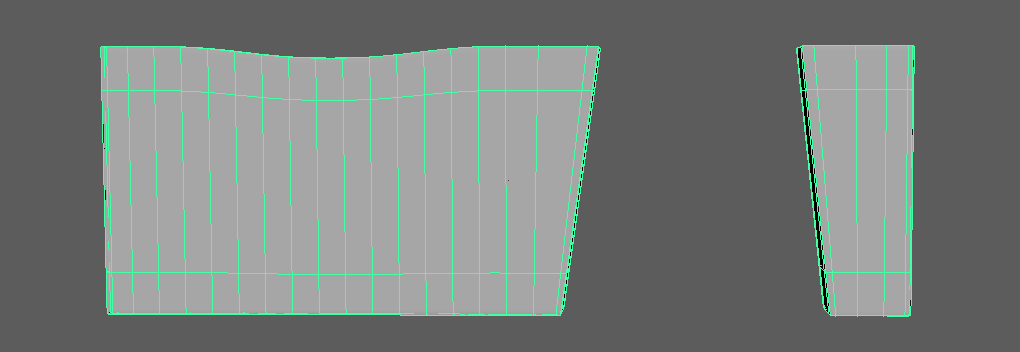
* Land Module 2



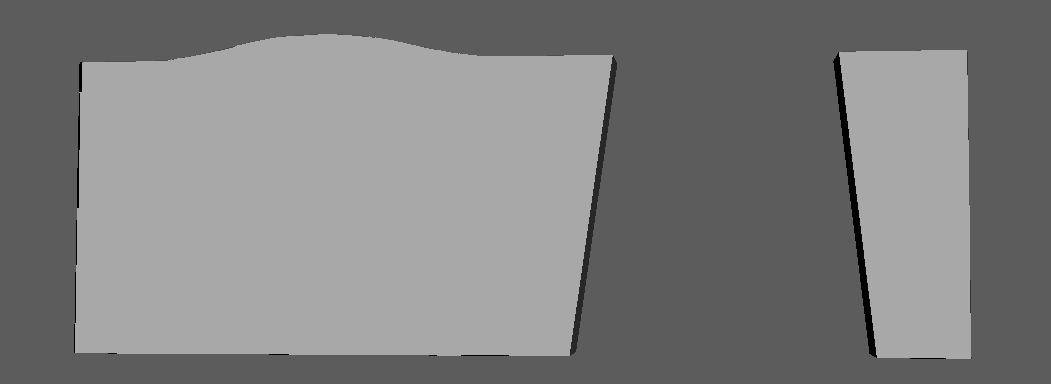
* Land Module 3



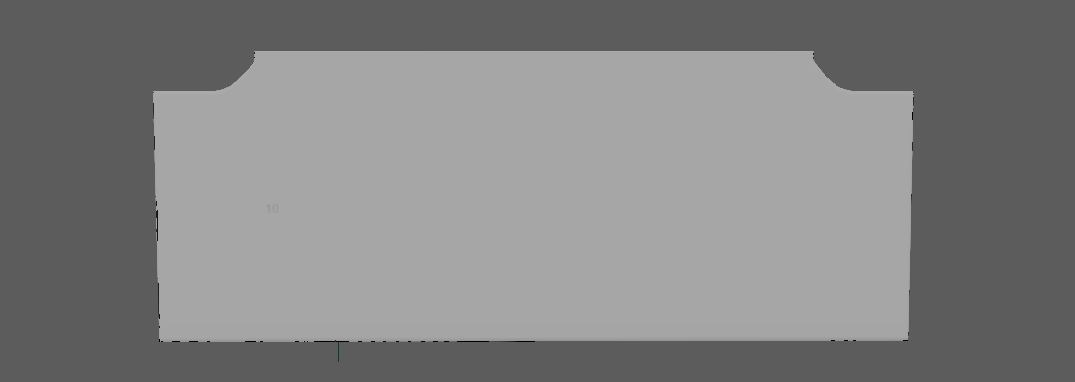
* Land Module 4



* Land Module 5



* Land Module 6



* Land Module 7



* Land Module 8



**~~Cinematics~~**

~~These are the 2D or 3D scenes often shown as an intro, between missions, and at the end of the game. These should be scripted like a film script as separate documents. This, however, is production work. For the purposes of the functional spec, just list them here with the general purpose, content and target length. If any video is involved, list it in the following subsection.~~

**~~Video~~**

~~Unless you are doing an FMV (full motion video) game, this subsection is pretty light. If you have any video in your GUI for say pilot messages, break it down here. All video tasks will require scripting, but that is production work. List the general purpose, expected length, and general content like number of actors and set design, even if it ends up being blue-screened into a 3D rendered background.~~

**~~Marketing and Packaging~~**

~~You might as well list it here and the schedule, because they’ll ask for it. This includes web page art, sell sheet design, demo splash screens, magazine adds, press art, the box and manual.~~

**Sound and Music**

**Overall Goals**

The overall goals of the sound effects are to evoke a sense of urgency and action to the player. We want sound effects that fit together and make sense to keep the players interest and add an interesting element to the game.

**Sound Effects**

SFX List

|  |  |  |
| --- | --- | --- |
| **GUI** | **Special Effects** | **Characters** |
| Button Clicks: File Name - ButtonClickSFX | Stars Hitting the Ground (Poofing): File Name - StarsHitGroundSFX | Wolf Colliding with Rabbit (whack): File Name - WolfCollisonSFX |
|  | Catching Stars (jingle): File Name - CollectingStarsSFX | Rabbit Collision with Objects (thud): File Name - ObjectCollisionSFX |
|  | Losing Stars (urgent jingle): File Name - LosingStarsSFX | Falling Into Chasm (Whistle): File Name - FallingSFX |
|  | Rabit Speed Boost (wooshing): File Name - SpeedBoostWindSFX | Hawk Colliding with Rabbit (Metal Clink): File Name - HawkCollisonSFX |

|  |  |  |
| --- | --- | --- |
| **Gameplay Elements** | **Terrain** | **Motion** |
| Wolf Howl Alert: File Name - WolfHowlAlertSFX | none | Rabbit Running (foot falls): File Name - RabitRunningSFX |
| Hawk Cry Alert: File Name - HawkCryAlertSFX |  | Rabbit Speed Boost (faster foot falls): File Name - RabitSpeedBoostRunningSFX |

|  |  |  |  |
| --- | --- | --- | --- |
| **Music** | **Events** | **Shell** | **Situation** |
| Only Music required for the game must be upbeat, suspenseful, and able to loop endlessly as this is an endless runner game. | Death: when the player dies a sad, funeral like song plays | Title Screen: an upbeat, but not suspense full song plays | none, since this is an endless runner. |
|  | Success: when the player reaches a certain amount of stars a bell like jingle sounds |  |  |