

Lee Yoerns

12 December 2024

CIS129

Chang

PROBLEM IDENTIFICATION

I am designing a game that needs more work done on the actual in-game features, rather than just the overall story.

- UI
- Logic of mechanics
- Fighting style/mechanics (+ animation)
- Sound design timed with certain events
- Character sprites and dialogue boxes

EXISTING SOLUTIONS

- Fighting Style: rhythm games
- Mechanics Logic: choose your own adventure games/visual novels with multiple endings
- Sound Design: Don't Starve has character voices based on specific instruments; Hourglass Echo's fighting system is music based, so maybe take inspiration from this idea? Battle themes use certain instruments/motifs based on who you are fighting with/against
- Character Sprites: visual novels, games with lots of character dialogue (ex: Stardew Valley)

QUESTIONS FOR SELF:

- What do I want the UI to look like?
 - How do I make sure it does not clash with the overall mood, but still keep it unique and interesting?
 - What is needed for Hourglass Echo's in-game UI?
 - How would the interface differ between overworld exploration, battle time, etc? (keep it similar so it's not too disorienting, but also make them different enough to make it clear to the player that they have different abilities/functions during battle time)
 - Pokemon

APPLICATION OF FEATURES:

- Character themes: when character is onscreen, play (insert looping sound file)
- You name your character at the beginning of the game; that input is saved so that characters can refer to it in their dialogue
- During battle time, hits you do within a certain millisecond timeframe of the note will do extra damage; the further away you get from being "on beat" with the music, the less damage you will do, and the more likely it is that your hit will miss. Additionally, when needing to dodge, being on beat will prevent you from taking any damage, and the further away you get from being in time will make you take more damage.
 - = Fighting gameplay is less based on hitbox and more on rhythm to the music; harder fights may have music with strange/irregular patterns, or tempo and/or time signature that changes in the middle of the song (ex: second boss stage)

- The story may change based on your choices (create a map to include in the presentation?)
 - However, there is a certain character that is coded with the idea of having “free will”; his choices are either RNG-based, and/or overall “big decisions” that can change the entire ending are based on the number rolled at the beginning of the game in an intro scene (dice game)
 - The game is based around a time loop, where the player must go through the same story again and again in hopes that they can finally break out of it. The only way it is possible to break free is if a “five” is rolled at the beginning of the game, which puts the aforementioned character on the path to do what will enable the player to get the “good true ending” (provided they also make the other choices necessary).
 - A “bad true ending” is also possible, although it is less based on the number rolled
 - This ending is achieved through helping the free-will character become a robot. If you succeed, the truth about the game will be revealed, simultaneously breaking the time loop and starting it again. You as the player will no longer be trapped in the time loop, and you will achieve a “game ending” rather than a “loop ending”. The character will still be trapped within the time loop
 - The player can go down the route of trying to help this character NOT become a robot; if they play their cards right, it is possible, but he will still die at the end no matter what your choices are, even if not by means of trying to replace his human body with robotics; this is because he is also trapped in the time loop, and is the reason for its existence in the first place.
-

- Certain choices the player makes are hard-coded to result in events going a certain way (e.g. saying you are a heavy sleeper means you will sleep through events that take place at night; one of the events that determines the difference between one of the “good loops” and the “bad loops” end comes down to being able to witness an event that occurs during the night)
 - Some choices that would otherwise result in the occurrence of a specific outcome can be overridden by other choices (in above example, player may still witness the nighttime event if they say they are a night owl)

PSEUDOCODE

Determining how much damage a player will take:

```

player_reacttime = user_input
#user_input is the amount of time in milliseconds between the
beat and the registered moment the player reacted
perfect_range = (0, 51)#range max is 2nd number -1
good_range = (51, 101)
ok_range = (101, 151)
bad_range = (151, 201)
#any input delayed more than 200 milliseconds is considered a
miss
if player_reacttime in perfect_range:
    dmg_taken = 0
elif player_reacttime in good_range:
    dmg_taken = user_input
elif player_reacttime in ok_range:
    dmg_taken = user_input + (user_input/2)
elif player_reacttime in bad_range:
    dmg_taken = user_input * 2

```

```
else:
    dmg_taken = 500
```

Damage will be reduced if a player is wearing defensive gear:

```
if decent_gear in equipped_items:
    #equipped_items is a list of gear, weapons, etc the player has
    #chosen to use
    total_dmg += (dmg_taken//1.5)
elif good_gear in equipped_items:
    total_dmg += (dmg_taken//2)
elif great_gear in equipped_items:
    total_dmg += (dmg_taken//3)
elif amazing_gear in equipped_items:
    total_dmg += (dmg_taken//4)
```

Sprites change based on player response.

- Class types for emotional implication of player choices (ex: good response, mean response, neutral response)
- If player_choice = .good character_sprite = [character_name]_goodsprite.png
 - Same for other responses

what does true_good do? Function? Needs to exist in the background; some things are the same regardless of dice roll, some events only occur with certain numbers

- Function exists to toggle events? Function of functions?

Each scene where the player makes a choice is a function. In each scene, certain events are possible depending on dice roll. Example: in a scene where the player has decided to hang out with the character in the time loop, the function might have a section such as:

```
    If truegood_possible():
        root_question()
```

#rest of code outside of if statement. If the statement is false, nothing else should trigger and the code should continue as normal.

- Maybe instead of function for truegood, rename dice roll = 5 to truegood, so would instead be if truegood: (= if dice roll is 5)

Determining the good true ending:

```
import random
d6 = [0, 1, 2, 3, 4, 5]
#5 is a symbolically important number in the world of Hourglass
Echo
dice_roll = random.choice(d6)
if dice_roll == 5:
    good_true = dice_roll
```

Potential application of the good true ending:

```
#Within an event (= within a function):
print('Dialogue Option 1')
print('Dialogue Option 2')
if good_true:
    print('Dialogue Option 3')
#The third option only appears for the user if the good true
ending is active, which only occurs if the roll at the beginning
of the game is a five.
```

Determining which character song will play:

```
examplechara_sonplist = [song1, song2, song3]
if mood = normal:
    song1
elif mood = sad:
    song2
else:
    song3

#each song variable leads to an audio file playing
#mood is a variable determined at the beginning of each event,
but has the potential to change based on the choices made within
an event. If the event begins as mood = normal, but the player
```

makes a choice that makes a character upset, the song for the sad mood would begin playing instead.

OPEN QUESTIONS:

- How to determine when a beat occurs in the audio?
- How would the “mood” variable actually be programmed?
 - Maybe with classes?