Owen Badgley
Literary Games
5/9/2024

Making a Game

Homeostasis: the tendency toward a relatively stable equilibrium between interdependent elements, especially as maintained by physiological processes

The Idea for the Game

When I was trying to think of an idea for a game that I wanted to make, I started out by thinking of what kinds of games I enjoy playing. First I thought of Souls-Likes, such as *Dark Souls* and *Elden Ring*, and then thought that a 3D Action RPG would be much too hard for my current game development experience. After that, I thought about making a metroidvania, such as *Hollow Knight* or the titular *Metroid*. While I think that given enough time I could eventually create a serviceable metroidvania, I didn't think that the amount of time I had to work on this project would be enough to finish that project. So, finally, I settled on something that I thought I could make: a 2D platformer.

After that, though, I realized that that was the thought process that every inexperienced game dev goes through, and because of that, there are more 2D platformers out there than could ever be imagined. After this realization, I retreated into my mind to try to think of some ideas that could make my game different from all the others. One of the ideas that first struck me, perhaps because we had just played *Thomas Was Alone*, was the idea of controlling multiple characters at once. I didn't

want to do it in the same way *Thomas Was Alone* did it, though. I wanted the player to control every character at once. I thought that this would be a very interesting premise for a puzzle platformer. The ideation process that lead me here can be seen in Appendix 1.

Quickly after trying to think of some level ideas with this premise, however, I realized that the levels for this game would have to be incredibly intricately crafted. I am not sure that as my first game, I should try to make a game that is so much of a game design challenge. So finally, I settled into the idea that instead of a puzzle platformer where one person controls two characters, I have two people that control one character each. This is where the idea for my game, *Homeostasis*, was born.

Homeostasis is a game where each player controls one emotion, and the end goal of each level is to get the two characters to meet, and then the players can move on to the next level. This is very loosely a metaphor for reconciling emotions. In an ideal world, if I had enough time and talent to make this game fully realized, there would be many different worlds. Each world would have two different emotions for the players to play as, each with a different movement ability, to keep the game fresh.

The art style that I went with is a sort of chalk on chalkboard style, with each of the characters drawn in a bright color chalk, and all of the levels drawn in white chalk. I did this because of the story I finally decided on. While the story didn't make it into my actual coded project, I did think of an idea for it, with not enough time to implement it.

The story would follow an elementary school teacher, hence the chalkboard theme. Each world would be about a different student, and the gameplay would be a metaphor for the dialogue that the teacher would be having with the student.

Due to the characters being emotions, the stories would all be related to the mental health of the students. The two characters that I have created so far, the little happy guy and the little sad guy, would be in the world where the teacher is talking to a kid who is feeling depressed. They would work through it together, and at the end of the world there would be a cutscene of some sort that shows the student and teacher together, and the student leaves with a smile. At the end of the game, after all of these worlds have been completed, and the stories are hopefully moving, there would be a last day of the school world. This would be where there are one or two levels for each pair of characters, and perhaps even the characters get jumbled up and different combinations get to shine. This is a sort of metaphor for a teacher who managed to get a classroom full of kids to open up and finally be able to talk about their emotions, something that is heavily stigmatized in grade school.

Back to the gameplay, while I was designing the game one night I went back and played a couple levels of a game that came out a couple years ago, *Neon White*. This is a game where you are trying to kill a bunch of demons in set levels as fast as possible. This game came to me at the perfect time, because I wanted my platformer to be hard, but accessible. The gimmick of *Neon White* is that there is a timer on every level, and if

you beat the level in a certain amount of time, you get different medals, and there are global leaderboards and all of that fun stuff. I think that what my game was missing was a sort of timer to improve on your times throughout each level. I don't think that the game is long enough to support being called more than a tech demo, so a timer would allow for a lot more replayability. I have also never seen a two player speedrun style game, so that could be either very fun or very frustrating. Perhaps there is a reason I have never seen one.

Challenge in *Homeostasis*

As for how this genre of game has been understood by critics, 2D platformers are an extremely wide range of genres. What I am going for, however, is something that I think is more specific than just a 2D platformer. I think that of the critics who we have read in class, Juul would have the most to say about my game. This is because I was striving to make a game that heavily emphasized failure, both in trying to beat the levels and trying to obtain a faster time on the levels. I believe that failure is one of the fundamental parts of games that makes them fun. Without failure, there is no inherent difference between a game and a movie. What a game can provide to the player is friction, a game gives the player something to struggle against if they want to see how the story plays out. As Juul said, "It is the threat of failure that gives us something to do in the first place." (Juul, 45)

Without that failure, playing a game can become a mindless activity. This is not a bad thing necessarily, but I was hoping to make a game that the players would have to fight with. Ideally they would have to work together to achieve the best possible time, although I didn't have the mechanics in this prototype to design levels that leveraged that feeling. Juul says "Much of the positive effect of failure comes from the fact that we can learn to escape from it, feeling more competent than we did before." (Juul, 45) He says this in reference to learning to overcome an obstacle in a video game. I think that this feeling of overcoming something that seemed previously insurmountable can conjure up some of the most visceral and memorable feelings in all of gaming, and I wanted to make a game that could hopefully capture some of that feeling.

I also chose this medium of a challenging two player game because I thought that it would be somewhat representative of the subject matter. I think that trying to work with a friend to beat a challenging level of a platformer could be analogous to working with yourself to try to figure out why you are feeling a certain way. Instead of having a conversation with yourself, you have to communicate with another person to solve a common goal. Talking to another person is also a very good way that someone can figure out why they are feeling a certain way. This is represented in the story of the game through the teacher talking to students, and helping them figure out why they are feeling the way they are feeling. It is hard to reconcile one's own emotions almost all of

the time, and I wanted to have gameplay that could evoke that feeling in the player. It is not easy to understand why something makes us feel the way that it does.

Design Challenges

In line with the game being one that is supposed to challenge the player, the creation of the game was something that repeatedly challenged me throughout the process in different ways. The first and most obvious way that I was challenged by this process was during the process of actually figuring out how to code the game. I have a lot of experience programming, but I had little to no experience creating a game before this project. This is the first game that I have ever made, and while I don't think it turned out perfect, I am honestly pretty happy with it for a first attempt. That doesn't mean that it was without challenges, though.

The first challenge that I had was figuring out how to code the grappling hook. I tried a few different methods on my own, and I could never get every part of it working. I figured out how to get a vector that was attached to a wall, but I couldn't figure out how to make the player swing around it. Eventually, I had to consult a youtube video on how to create it, which I referenced in the comments of the code. I didn't copy the code from the video one to one, I just used it as a reference for the right way of thinking about creating a grappling hook. If there is one thing that I have learned throughout my programming experience, just looking something up is almost always the best way to solve a problem. The code for this can be seen in Appendix 2.

The second programming challenge I had was creating a game that had more than just one level. I had a place where I could jump around and test out the movement, but I wasn't exactly sure how to program a scene switcher or any of that fun stuff. For this one, after the grappling hook fiasco, I went to youtube immediately, and found a very good tutorial for implementing exactly what I needed. This one is also cited in the code, but it had some bugs that I had to work out, because the tutorial was made for a version of Godot that was before version 4, and it was somewhat broken in version 4. The code for this can be seen in the Appendix 3.

Another challenge was creating levels. This was mostly due to my own fault, because I spent so much time creating all of the core mechanics that I would need to pump out dozens of levels, but I gave myself barely any time to make actually good levels. I think that given more time, I have built quite a promising base for a game, I just didn't have time to realize that base into anything more than just a tech demo for what I had created. I made a tilemap, I made everything very scalable, I just didn't have time to build it at the scale that I envisioned for it. This led to a lot of the levels that I created feeling kind of repetitive, derivative, and just generally creatively bankrupt. This was by far the biggest disappointment of the project for me, because I thought that I would be able to make good levels at a rate that was much faster than what I am actually capable of.

The Future of *Homeostasis*

Given everything that I said in the previous sections, I think it is pretty clear what I want from a finished product. I want something that has more levels, an actual story built into the game, and much better levels than the ones that I have currently. Once again, I think that I have built a very good base for a game that could be extremely fun if properly realized. I think the idea of a *Neon White* style two player platformer could lead to some incredibly fun moments and gameplay.

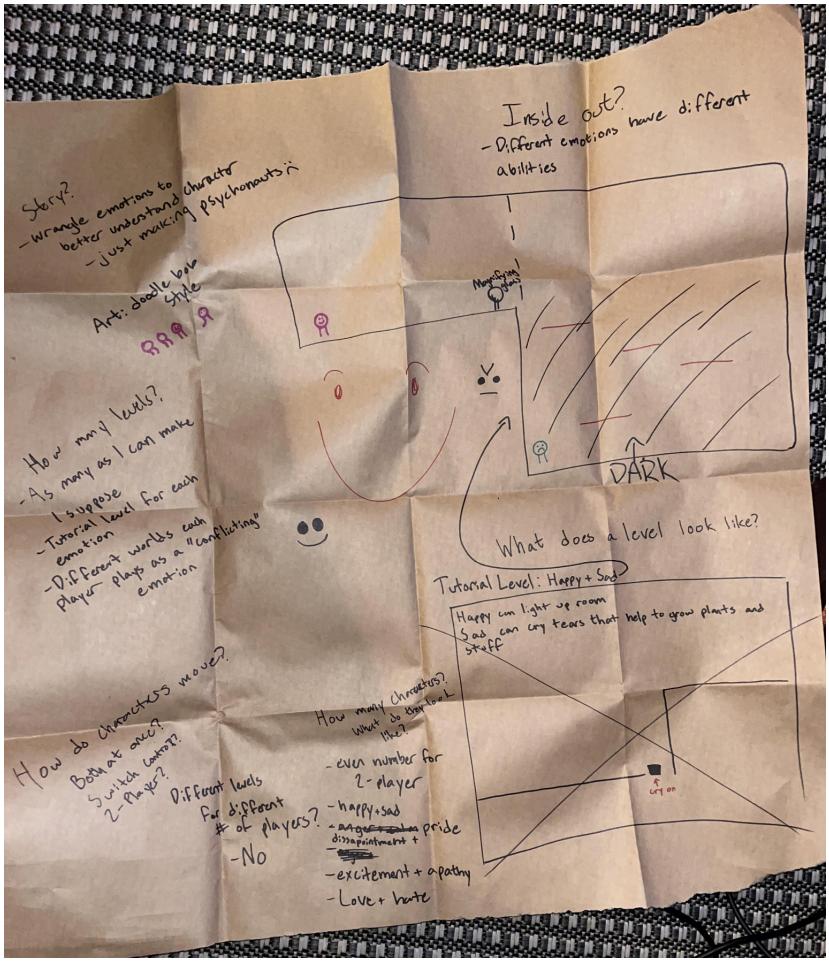
I think that this speedrun style approach is what could give this game much more open-ended play. I think that the addition of a timer makes it so that those who want to beat all of the times, and unlock all of the medals, have a much deeper game that they can enjoy than those who just want to play to see the story. I think this idea is pretty neatly summed up by my favorite definition of a video game that we have looked at this semester. It hooked me from the very first day, and it is the quote from Bernard Suit that says “Playing a game is the voluntary attempt to overcome unnecessary obstacles.” (Suits 54-55) I think this perfectly sums up why I love games, and I think that for this game, and any game I make in the future, because I intend to make games in the future, this quote is one that I will keep in the front of my mind throughout the entire process.

Bibliography

Suits, Bernard. *The Grasshopper: Games, life and utopia*. 1978, ci.nii.ac.jp/ncid/BA88557439.

Juul, Jesper. "The Art of Failure: An Essay on the Pain of Playing Video Games." *Choice/Choice Reviews*, vol. 51, no. 03, Oct. 2013, pp. 51–1301. https://doi.org/10.5860/choice.51-1301.

Appendices



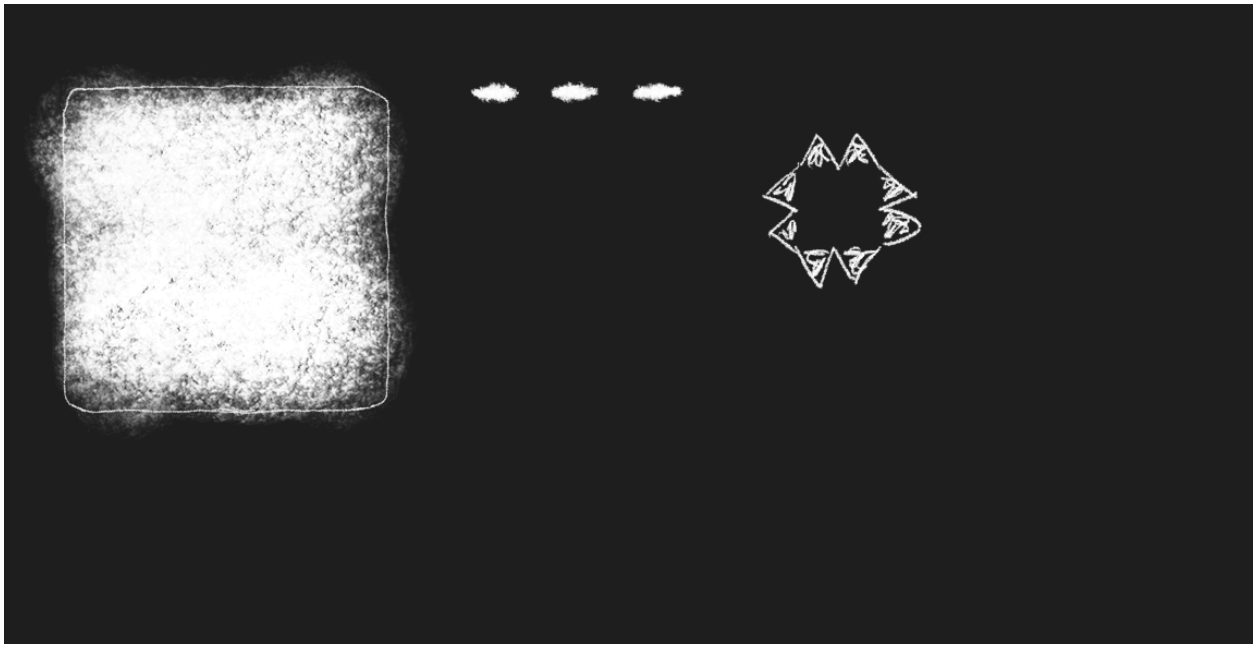
1. Paper Prototype

```
var hooked = false

func shoot(dir: Vector2) -> void:
	direction = dir.normalized()
	flying = true
	tip = self.global_position

func release() -> void:
	flying = false
	hooked = false


# Called when the node enters the scene tree for the first time.
func _ready():
	chain = $chain
	pass # Replace with function body.


# Called every frame. 'delta' is the elapsed time since the previous frame.
func _process(delta):
	self.visible = flying or hooked
	if not self.visible:
		return
	var tip_loc = to_local(tip)
	chain.rotation = self.position.angle_to_point(tip_loc) - deg_to_rad(90)
	$tip.rotation = self.position.angle_to_point(tip_loc) - deg_to_rad(90)
	chain.position = tip_loc
	chain.region_rect.size.y = tip_loc.length()
	if tip_loc.length() > length:
		release()

func _physics_process(delta):
	$tip.global_position = tip
	if flying:
		if $tip.move_and_collide(direction * SPEED):
			hooked = true
			flying = false
	tip = $tip.global_position
```

2. Grapple Hook Code

```gdscript
@onready var current_level = $Main
# Called when the node enters the scene tree for the first time.
#https://www.youtube.com/watch?v=XHbrKdsZrxY&t=1129s <- the video that helped me implement this
func _ready():
	current_level.level_changed.connect(handle_level_changed) # Replace with function body.
	current_level.restart_level.connect(handle_restart_level)

func handle_level_changed(current_level_name: String):
	var next_level
	var next_level_name: String

	match current_level_name:
		"main":
			next_level_name = "level_1"
		"level_1":
			next_level_name = "level_2"
		"level_2":
			next_level_name = "level_3"
		"level_3":
			next_level_name = "level_4"
		"level_4":
			next_level_name = "level_5"
		"level_5":
			next_level_name = "level_6"

	var temp = load("res://levels/"+next_level_name+".tscn") # res://levels/level_1.tscn
	next_level = temp.instantiate() # this is required in order to see the scene
	call_deferred("add_child",next_level) # call_differred fixes a few errors
	next_level.connect("level_changed", handle_level_changed)
	next_level.connect("restart_level", handle_restart_level) # syntax is a little different in version 4
	current_level.queue_free()
	current_level = next_level

func handle_restart_level(current_level_name: String):
	var next_level
	var next_level_name: String

	next_level_name = current_level_name

	var temp = load("res://levels/"+next_level_name+".tscn") # res://levels/level_1.tscn
	next_level = temp.instantiate() # this is required in order to see the scene
	call_deferred("add_child",next_level) # call_differred fixes a few errors
	next_level.connect("level_changed", handle_level_changed)
	next_level.connect("restart_level", handle_restart_level) # syntax is a little different in version 4
	current_level.queue_free()
	current_level = next_level
```

3. Scene Switcher Code



4. Level Assets

5. Character Sprites (Unanimated 😕)