

CS Capstone Planning Document

Defining the Problem

Educational games have a unique opportunity to incite excitement and a genuine desire to learn about a subject in players, transforming learning or skill application from a chore into play. Inspired by such games as picoCTF 2019, Terminus, and TwilioQuest, I have decided to develop a calculus themed game that provides calculus students an opportunity to learn and practice BC calculus concepts through the frame of an interactive story and fun gameplay. While the game is designed with current students of calculus in mind, it will also provide opportunities for players new to calculus to learn on their own, and allow players who have taken calculus in the past to refresh their skills. The goal is for the game to be fun, improve player's calculus skills, and inspire players to enjoy calculus more out-of-game.

The main client will be students of BC calculus, such as Miss Kiersten Schmitt, who took calculus AB as a junior and is self-studying calculus BC this year. As recommended by Miss Schmitt, core features that would make this game most engaging and enjoyable include: an engaging storyline, calculus problem generation and solving that is integrated into the story, problem difficulty that scales with the player, and fun art and music. While such a game doesn't solve a 'problem' per say, it does provide a service as fun tool for practicing calculus. Miss Schmitt indicated that she would use such a game as an extra study tool for calculus, and just for fun as well.

Rationale for Proposed Product

My proposed product is as follows: a 2D RPG, working name “The Legend of Calculus”, in which the player navigates a fantasy-themed game world on their quest to collect the 10 “pebbles of calculus” and find a way home to their own world. The player goes on quests for historical mathematicians who contributed to calculus, completing a mixture of purely gameplay focused quests and calculus based quests that require the player to solve calculus problems in order to complete. As the player solves calculus problems and earns XP, new areas of the world unlock, each one (and the quests/problems within it) themed after a unit of BC calculus, and new pieces of the story are revealed. Each level/room would have its own contained mini-story that would factor back into the overarching story of the player attempting to return home at the end. The core feature of the game is solving the calculus problems, which have several variations: quick problems that can be solved in under two minutes which will be integrated into a combat system with enemies; longer problems (similar to FRQs) that must be solved before entering a new map area or during a final boss fight/end of level quest.

Such a product would meet much of the criteria the client Miss Schmitt specified in our interview. For example, she said “I think it would be fun if you incorporated any of the calculus history- I would be an absolute nerd for that”: a storyline focused on historical figures from calculus would be engaging for her and other calculus students curious about the history of calculus, and provide motivation to continue solving the problems and advancing the game. Having rooms focused on individual units of calculus that got harder as the player progressed would also meet her request of having challenges that scaled with the player, as she said she liked other learning based games that had a “good balance between the [challenges] and gameplay at the start, but then as the levels got harder there was more learning.” Additionally, potentially adding an option to track player streaks would encourage the client to use the game to practice calculus every day, as she is a “streak fiend” (however that would be an enable/disable option, as some people don’t like having streaks tracked).

The user will interact with the product by playing the game: they will be able to navigate their avatar through the game world, collect items, go on quests, fight monsters, enter answers to calculus problems, earn XP, and progress through the story.

The software necessary to complete the project includes the following: Unity as the main game engine, Visual Studio Code with .NET extensions as the code editor, C# as the main scripting language, Python as the mathematical backend for generating calculus problems and solutions, Tiled as the map-designing software, and Aesprite as the pixel art editor for sprite creation.

Success Criteria

The goal is to create a viable product akin to a Beta, that the client can play and enjoy, even if every single room/map area or the whole story is not necessarily finished and polished. All main systems (quest, dialogue, inventory) and core mechanics (movement, problem solving, story progression) will be finished and fully functional, and will have a user friendly UI that the client can interact with to see details about their gameplay. I've laid out several stages of the game below and what criteria I would like to meet in each stage. The goal of the final project would be to make it to the "Substantial Viable Product" stage, but I added in some criteria for after that too for if I get ahead of schedule and have the ability to do more than I currently think I will.

Minimum Viable Product 1 (The Very Basics/Main features without fluff):

Goal to Finish: 3 - 4 weeks

1. Player Movement
2. Turn based combat
 - a. Calculus problem generation
 - b. User UI displays problem and answer textbox
 - c. Enemy damaged when problem answered correctly
 - d. Timer determines how much damage player does
 - e. Defense requires answer in under time limit to avoid player damage
 - f. Player death respawns at level "safe zone" with copy of inventory and backed up progress from before fight started
 - g. Player victory awards XP or item
3. Settings Page
 - a. User can open settings UI and quit game
4. Dialogue System
 - a. Players can initiate dialogue with NPC
 - b. NPC dialogue can be updated after quest completion/story event
5. Quest System

- a. Player can start quest by talking to NPC and selecting certain dialogue option
 - b. Quest system tracks quest requirements.
 - i. Support for item collection quests
 - ii. Support for combat quests
 - c. Player is rewarded with XP/item for quest completion
 - d. NPC dialogue reflects different quest stages of completion.
6. Inventory System
- a. Player can pull up inventory UI by pressing I
 - b. Player can add items to inventory by colliding with them in the game world or when quests are completed
 - c. Items can be subtracted from inventory
7. One room that branches off of the main area

Slightly-More Substantial Viable Product (Starting to add more features that enhance experience)

Goal to Finish: 3 weeks

- 1. Settings Page
 - a. User can quit game, save game, adjust volume controls, see help page with controls
 - b. When settings are open the game is paused- players and enemies cannot move.
- 2. Quest System/Player Journal
 - a. By pressing Q player can open Journal UI, which keeps track of open quests and their criteria in a player friendly format as well as provides character insight into the story.
- 3. Room Completion
 - a. One room is flushed out functionally, with NPCs that require multiple quests to be completed.
 - b. Upon completion of enough quests in room player can unlock final quest in room
 - c. This “final quest” will be solving some sort of FRQ problem
 - d. Reuse combat calculus problem answer, but no time limit (or much longer time limit, like 10 mins)
 - e. Completing Room adds pebble to user inventory
- 4. Story Beats
 - a. Game can initiate dialogue without user prompt for story “cutscenes”

- b. Main door can be unlocked and game can “finish”
- 5. Save state
 - a. Player can save game and restart with their progress intact.

Substantial Viable Product (AKA An Enjoyable Beta and the MAIN GOAL)

Goal to Finish: 2 weeks

- 1. Start Screen
 - a. When game starts title card and start button, options button.
- 2. Story Beats
 - a. Story is written and has beats the player can hit
 - i. Start, each collection of a pebble initiates a story cutscene/dialogue related to story, end when door unlocked
 - b. Story knows when certain checkpoints have been reached and triggers dialogue or new main story quest
- 3. Room Completion
 - a. 5 of the 10 rooms have quests and can be fully played
 - b. NPCs have dialogue
 - c. Calculus problems and content have been reviewed and approved by clients
- 4. Art and Animations
 - a. Main character has a pixel art sprite
 - i. Idle, running, fighting, hurt, surprised, happy, angry animations
 - b. NPCs have pixel art sprites
 - i. Idle animations
 - c. Enemies have pixel art sprites
 - i. Idle, running, fighting, hurt animations
 - d. World has pixel art tileset with levels, boundaries, and collisions
 - e. Each room has a different looping soundtrack

Complete Product (AKA I somehow pulled it all together and start adding polish)

Goal to Finish: 2 weeks

1. Room completion
 - a. All 10 rooms have been fleshed out with pixel art and have unique designs
 - b. Each room has different types of calculus problems in it fitting their corresponding unit.
 - c. Each NPC has 2 calculus related quests and 1 historical quest for player. 2 - 3 NPCs per room.
 - d. All 10 pebbles can be collected
 - e. Extra details, items, and areas have been added to the room.
2. UI Refinement
 - a. Settings, Journal, and Inventory UIs are all refined and have a pleasing, functional, and easy to use design
3. Character Customization
 - a. At the start of the game player can change element's of sprite customization like name, hair, hair color, skin color, shirt color, etc.
4. End screen
 - a. Upon completion of the game story, there are credits for any assets and music used
 - b. Reopens to post-game where player can continue to explore and do side quests

Refined Product (AKA a miracle occurred, I somehow finished everything else, now I'm just challenging myself)

1. Economy
 - a. There are shops in the game world that sell outfits and equipment.
 - b. Certain outfits/weapons have different perks in combat
2. Unique Room features
 - a. Graphing feature for the polar equations room
3. Website
 - a. Game is hosted online and users can play it in an embedded form.

Appendix A: Interview with Client

Me: Okay. Hello Kiersten! Can you introduce yourself?

Kiersten: Hi I'm Kiersten. I'm a senior in high school. Calculus enthusiast I suppose ;)

Me: Ok awesome. So, first question: how often do you study calculus outside of homework, before tests... oh wait I guess you don't actually have homework [NOTE: Kiersten self studies calculus during 8th period, is going to take the AP exam, but isn't in a formal class]

Kiersten: Yeah, so back when I was in a class it was usually just before tests and a little bit before quizzes. Nowadays I use online software so I do a little bit every day. That way I can keep up my streak. I'm a streak fiend, so I have like a 30 day streak. So yeah, either in bursts before tests or a tiny bit every day just to keep up my streak.

Caroline: You said you use online software. Can you explain what type of software you use and how it works?

Kiersten: So, the first time I look at [the calculus material] on Khan Academy, then I do practice problems from an old AP calc BC workbook. Then I go over it with Brilliant. It goes over the same content, just with a more nuanced approach. So I like using that because it like, reaffirms it.

Caroline: Okay cool. So have you ever played a learning based game in the past? Where the game story focuses around learning a certain type of concept?

Kiersten: Umm... I don't think so? Not a huge gamer. Uh, I don't have a good answer for this question.

Caroline: It's okay. If you haven't played one that's an answer.

Kiersten: okay cool

Caroline: But if you WERE ever to play one, and it was based on calculus, do you think you would use it?

Kiersten: Yeah I think so, especially if my friends played it.

Caroline: What types of features do you think would make it enjoyable? What would you like to see?

Kiersten: I think- big fan of- hey do you remember those old picoCTF games?

Caroline: YES. Hey you *have* played a learning based game.

Kiersten: OH YEAH! That wasn't the first thing that came to mind, but yeah. I liked how there was fun visuals, and music- and the lava map had the BEST music I'm sorry. But then like, you also had these quick problems that weren't super hard. At the start at least. It definitely got harder as you went on. I liked how you had the mini-quests and short problems, and how after each one you got to see more of the story or go to a new section of the map. And then there was a

good balance between the [CTF challenges] and gameplay at the start, but then as the levels got harder there was more learning. I liked that.

Caroline: So start off with easy things, a good story, and then as it goes on make the [learning] harder and [the players] are invested enough in the story to keep doing it.

Kiersten: Yeah! 'Cause I still remember 'A', and being lost-

Caroline: -hey wait did you ever finish it?

Kiersten: I did not ever finish it. I never won. But I wanted to know how it ended so I made you spoil it for me.

Caroline: Oh yeah. Okay, we'll speed run this last one. Do you have any more specific feature or story ideas?

Kiersten: Ooooooh, I think the history of calculus is very fun. It's a very silly history. I think it would be fun if you incorporated any of the calculus history- I would be an absolute nerd for that- I would be like *oh my gosh- it's Issac Newton moment?! Invention of the x-y coordinate plane moment!?* That's my hot take. I feel like people over hype up the invention of calculus, but they did not hype up the invention of the x-y coordinate plane as much as they should have. That was crazy- but not the point [unintentional pun?!?]. Go back to the questions.

Caroline: Okay last one. Any recommendations/requests for artstyle, music, gameplay feel?

Kiersten: Okay, I mean I'm a big fan of the classic, pixel art with 8-bit soundtrack.

Caroline: Yeah that's about all you're probably going to get out of me anyway.

Kiersten: Well yeah I mean what are you going to do? You could make yourself go insane over the art but do you WANT to do that? Anyways, if I have any more ideas I'll text you!

****music starts playing over speakers****

Caroline: Okay thank you so much! Bye!

Kiersten: Bye!

Appendix B: Data Structure Brainstorming

1. Inventory System:

- a. will probably be a an array of dictionaries? Store name of item, sprite representation, quantity in inventory.
- b. AddToInventory(string itemName) method
 - i. Check to see if item already in inventory
 - ii. If so inventory[indexOfItem][quantity] += 1
 - iii. Else append {name: itemName, quantity: 1, sprite:gameObject.Sprite}
 - iv. Called when player collides with collectible
 - v. Called when quest finishes that awards item (like pebbles)
- c. RemoveFromInventory(string itemName) method
 - i. Find index of itemName in array
 - ii. Remove dictionary
- d. Display UI
 - i. This will be the hardest part here... I have never done this before.
 - ii. Will create UI squares as scriptable objects? Attach script to inventory UI as a whole that runs through inventory array, runs through an array of UI “slots”, grabs their position on the screen, renders the sprite from the inventory object dictionary to the middle of that position?
- e. Extra: Mouse over for details
 - i. Extra key corresponds to text that will be displayed when User scrolls over square (so this script will need to be attached to the UI, not the inventory, but will still need to access data in the inventory data structure....)

2. Quest Manager

- a. Probably nested arrays
- b. BeginNewQuest()
- c. UpdateQuestStatus()
- d. CompleteQuest()

- i. Can call `AddToInventory()` or `RemoveFromInventory()`
- e. Display Journal UI
 - i. A bit easier I think since this will just be text...
 - ii. Scroll overflow
 - iii. Each new quest has a heading
 - iv. Criteria in bullet points under it
- f. Somehow needs to interface with Dialogue manager...
- g. Types of quests:
 - i. Collection quests (collect series of objects from game world)
 - ii. Combat quests (defeat a certain number of monsters/solve certain number of problems)
 - iii. Location quests (discover a certain location)

3. Dialogue Manager

- a. Each character sprite will probably have a dialogue property
 - i. This will probably be an array containing basic lines to cycle through
 - ii. Actually there might be several arrays- one for general dialogue, one for each quest the NPC can give, one for post-room-completion dialogue, etc.
 - iii. Only one array at a time will be set to “active dialogue” variable
- b. `SpeakWithNPC()` will access `currentDialogue` array and step through the lines.