# Research and Problem Selection

### Identify a Problem :

I’d like to be able to capture short clips of things that have just taken place, rather than having to have the foresight to record it and then scrub the footage for my moment of choice.

### Explore Existing solutions:

* **Nvidia Shadowplay** : Works, but comes bundled with software for your Nvidia graphics card, can be slimmed down to reduce bloat.
* **Bandicam** : Works, but requires premium for videos without a watermark.
* **Medal.tv** : great for recording video game highlights, but also requires premium for most features
* **OBS Studio** : Feature rich and not locked behind premium, but support barely exists and it’s very hard to learn.

### Justify your choice

I’ve been using my computer most of my life now and I cannot count the number of times that something significant has happened and I find myself saying “I wish I had been recording that !” This is only a minor inconvenience, but if your workflow on your personal computer is fast paced, and you’re liable to miss something if you aren’t paying attention, having a software to record it for you would be nice.

# Design a Solution:

### Propose a software solution:

I’d like my software to have forks compatible with any operating system, including Windows, Linux, and macOS. I’d also like this software to be very minimal and lightweight. No added configurations on in-game filters or network analytics or anything excessive. I’d like the software to have a customizable length and quality for the clips that are taken, and i’d like it to not require an entire window’s worth of space(i.e I want it to be maybe 800 pixels wide at most, to keep it from being too distracting to open in the middle of an intense or time-sensitive situation.

### Pseudocode:

(Adding this note here, the components inside of video-clipping software are way above the level of programming that I’ve been taught so far, and If I were to make accurate pseudocode of this hypothetical program im making, I’d have to spend days each reading up on the intricacies of API’s, hardware acceleration, Multithreading, Encoding, Real-time processing, and all sorts of stuff that i’m not quite smart enough to understand just yet, so i’ll be including these in my pseudocode, but just very simplified.)

Define class “Main”

Import Modules/APIs # *Screen / Audio capture, encoding, etc.*

Initialize check for hardware/software compatibility issues.

If incompatabilitiesDetected = True  
 Display “Error! Incompatible {error1} detected!” && Exit

Initialize screen capture module

Framerate = userFrame

Codec = userCodec

Initialize audio capture module

sampleRate = userSampRate

Initialize encoding module

realtimeEncoding = True

Initialize emptyBuffer1

Enable video storage

Enable audio storage

timeLimit = userTimeLim # Will cut off recording at selected length.

Establish “while” loop() # *Where the program will run continuously*

Capture frame

Capture audio

Output > emptyBuffer1

If emptyBuffer1 > timeLimit

Delete oldestData # To maintain buffer size & prevent overflow

If userHotkey pressed:  
 Save data from emptyBuffer1

Output > userChosenDir, userChosenFormat

If userHotkey pressed > 5 in 1 second OR bufferOverflow = True:

Display warning message

Clear emptyBuffer1

Save mostRecentClip

Exit “while” loop  
Stop encodingProcess

Release openFiles && memory

### User interaction:



This is a very simple graphic design but it shows off a light color scheme and a simplistic UI. There will also be a settings UI where a user can define their desired clip length, output directory, video format, quality, etc.