**Timer Fundamentals**

Timer zone detection should be done with entity outputs (ie using triggers in the map to fire an output that the timer can read). Triggers in the map can be added using some sort of model that can be expanded and added in proper.

OnEndTouch ->

Start Zone - Timer for Map starts

Stage/Checkpoint Zone - Individual stage/checkpoint initial value recorded

OnStartTouch ->

Stage/Checkpoint Zone - Time stored from the initial time value is subtracted from the time when entering this trigger and recorded as the player's personal checkpoint record if it is faster than their last personal checkpoint record.

The timer prints the +/- in the hud for both the stage time and the overall map time. Points are given if it is a stage the player has not completed, and the results are printed in chat.

End Zone - Final time value recorded and checked if it is the players fastest/where it ranks among every other time for the map. Points are given and the results are printed in chat.

We will be able to add triggers into maps using StripperSource.

**Setting up Maps**

Because maps will use entity inputs and outputs, maps will likely have to be entered manually as text into the database. I'll leave the programming up to you, but I would suggest making map a class with the following parameters and appropriate assessors/modifiers:

Name - the name of the map

Tier - a value that describes the difficulty of the map decided by the person setting up the map, ranges from 1-6.

Type of map - Linear or Staged

Number of Checkpoints/Stages

Allow the person who sets up the map to create stage/checkpoint objects (they can be represented by a single object) with the following parameters (again, a stage would be best organized as a class):

Type - Stage or Checkpoint

Zone Type - Start Zone, End Zone, or Checkpoint zone

Name - The name of the Stage/Checkpoint

Index Number - a value ranging from 1 to the number of stages + 1.

Start Zones will automatically be assigned index number 1

End Zones will automatically be assigned the index number: number of stages +1

The person who sets up the map can set the stage index number manually (stage 2 would be set to 2, etc).

Entity Identification Number - the trigger entity's reference number built into the map file

**Points system**

Map Completed:

Points for completion of map = 10 \* [Tier]

Each stage of a Tier X map gives you X points, and when you complete the map it gives you what is left over to reach a X \* 10 total.

Example:

Tier three map with 6 stages.

Beat stages 1-6 : 3 pts per stage \* 6 stages = 18 points.

Completion bonus points = Tier \*10 - Tier \* # of Stages = 3\* 10 - 3\*6 = 30 - 18 = 12

Maps with more than 10 stages will simply give [Tier]\*[# Stages] + 5 completion bonus points.

Top10:

We took ksf’s formula using eureqa, there are less intensive but less accurate equations available if needed.

Note - I'll probably need to run a match with more data points to get the more exact formula, be ready to modify this.

Top10Points = 12.3449444536758 + 42.8979982526926\*cos(4.16349053295859 + 14.3961297722283\*[Total # of players with pr])/(Tier\*Rank) + (0.257081149204136\*[Total # of players with pr] + Tier\*Tier + 0.186297926759511\*Tier\*[Total # of players with pr] + cos(Tier) - 14.1390486230242)/Rank

Nofail Run - A Nofail Run can be detected if a player does not enter the same stage trigger more than once.

Additional Points = Tier \* 5

**Database Structure**

The timer will have to call upon massive amounts of information, so even a split second of lag when a player joins as the code searches the database for that player is bad. The database structure is probably the most important thing.