This document is designed to be utilized with the code engine which has the comment, ”This is almost ready” at the top of the code.

Disclaimer: There will probably be a lot of spelling/grammar issues. I am not proofreading this document. =) If you have questions ask. My cell number is (210) 716-1818

1. The idea of the game engine is a map allows the user to traverse through your scenario. The map is comprised of tiles and the tiles are each “square” the user interacts with the scenario.
2. There are two types of tiles: Mandatory tiles and fluff tiles. When the code builds the map, it places one copy of each of your mandatory tiles and then fill the grid with “fluff tiles.” Fluff tiles will be duplicated.
3. Mandatory and fluff tiles can have “event tags” (even though I recommend against fluff tiles having events). The event tags allow you to write code to test for them. (Look at my dynamite code) Soon as the user finds the lady in the boat, a random amount of moves later a dynamite explosion will go off. (I set the event when the hit the tile and I test for it later to “explode the dynamite.”
4. Don’t put items on fluff tiles. Make a list of items you want to put on the fluff tiles. The GenerateMap(Len,Width,MandatoryTiles,FluffTiles,AvailableItems) code will do it. The AvailableItems parameter is of list type.
5. Enough with the overview…now for the specifics! =)

The nuts and the bolts of your scenario …

1. I am after two “blocks of code” from you (if you want to go wild and do a lot more, or need engine code or classes changed, let me know!
   1. I need your scenario function (mine is called “RiverScenario()”)
   2. I need your variable/setup block of tiles and map information. In the code it is identified with:

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* River specific code start \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

(all of the code in between here I need)

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* River specific code end \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1. Let’s start with the “variable/setup block of tiles and map information” area first:
   1. Copy my code (might as well keep the “code start” and “code end” lines too
   2. Every line needs to be changed! Change my “River” to something specific to your code (Like Bay, Casino, Gulf)
   3. Compile your code and see if you get an errors (if so, fix them)
   4. The first three lines are variables (except for changing the river name, you can skip these lines)
   5. Change the {yournamehere}AvailableItems variable to have items unique to your area.
      1. Ex: CasinoAvailableItems=["$20 Chip","Empty drink glass","waitress number"]
      2. You can only put as many items as you have “fluff tiles.” Example, a 10x10 map with 9 mandatory tiles has 91 fluff tiles. Please don’t add more than your fluff tiles. It will break the code (I would prefer not to have to write code for this)
   6. Now make mandatory tiles … These are actual tiles you want the user to be able to interact with. (Maybe it just has an item they can pick up or just something they notice.)
      1. You have a variable called {yournamehere}Tiles (example RiverTiles) which is a list of mandatory tiles. This variable will hold all of your mandatory tiles. In my sample code, I have seven mandatory tiles, “”a pool of floating fish,” “someone fishing in a boat,” “River Tile 2,” etc … delete my tiles and make your own. The command to do this is:

{yournamehere}Tiles.append(Tiles.Tile("something here you want the user to see",1,"event tag","an item they can find"))

My example: RiverTiles.append(Tiles.Tile("a pool of floating fish",1,"","a nearly-dead fish"))

* + 1. Events and items are optional parameters. The first parameter is what is printed when the “enter” the tile. It is prepended with “You see “ when typed out.
  1. Next are fluff tiles. Same as above, but don’t put items. You can have events, but it could get ugly (because keep in mind you will have multiple copies of each “fluff” tile.
  2. Next is to run the GenerateMap() function to build your scenario map!
     1. It takes five parameters: Height in tiles (integer), Width in tiles (integer), your mandatory tile variable, your fluff tile variable, and a list of items you want added to your scenario.

Ex: BayMap=GenerateMap(10,15,BayTiles,BayFluffTiles,BayAvailableItems)

Above would create a 10x15 map and assign it to the “BayMap” variable.

* 1. Next I need the map actually be “converted” to a class (sort of). Change this line:

RiverMap=Maps.Map(RiverMap,5,5,"The river bank is too steep to exit here!") to your version. It accepts 4 parameters. The first three you basically need to “duplicate” from above. The first parameter needs to be your map variable. The next two are the same you put in earlier: Height in tiles (integer), Width in tiles (integer). The last is a string you want the user to see when you when they try to go off a map. For example I used, “The bank is too steep to exit here.”

* 1. I need you to add your items to the “master list of items.” I wrote a routine to read your map, so basically just call it and change it with your map name.
     1. Ex: MasterListOfItems+=BayMap.GetItems()