Here, I actually put a minute into this example, it will be a lot more helpful to you I'm sure.

```cpp
class MyScript : public ScriptImpClass {
     int playerid[127];
     int lastid;
 void Created(GameObject *o);
 void Enter(GameObject *o, GameObject *e);
 void Timer_Expired(GameObject *obj,int number);
 void Register_Auto_Save_Variables();
};

void MyScript::Created(GameObject *o)
{
 lastid = 0;
 for (int x = 0;x < 127;x++)
 {
  playerid[x] = 0;
 }
}

void MyScript::Enter(GameObject *o, GameObject *e)
{
   int end = 0;
   for (int x = 0;x < 128;x++)
   {
     if (playerid[x] == Commands->Get_ID(e);
     {
        end = 1;
     }
   }
   for (int x = lastid;x < 128 && end == 0;x++)
   {
     if (playerid[x] == 0)
     {
        playerid[x] = Get_ID(e);
        lastid = x;
        end = 1;
        Commands->Start_Timer(o, this, 123, 10);
     }
     if (x == 127)
     {
        x = 0;
     }
```

```cpp
    }
}

void MyScript::Timer_Expired(GameObject *obj,int number)
{
  if(number == 123)
  {
    int startarray = lastid + 1;
    if (startarray > 127)
    {
        startarray = 0;
    }
    for (int x = startarray,end = 0;x < 128 && end == 0;x++)
    {

      if (playerid[x] != 0)
      {
          GameObject *enterer = Commands->Find_Object(playerid[x]);
          playerid[x] = 0;
          end = 1;
      }
      if (x == lastid)
      {
          end = 1;
      }
      if (x == 127)
      {
          x = 0;
      }
    }
  }
}


void MyScript::Register_Auto_Save_Variables()
{
 Auto_Save_Variable(1,4,&playerid);
 Auto_Save_Variable(1,4,&lastid);
}
```

Lol, forgot the whole point of all my code, make it check if there is already a timer running for your soldier, lol, with out that check you may as well make the timer send the ID of the soldier as the number and then just have it get the id's when expired by the number lol.