

Rogue Elements

//Devlog

aug 14th 2017; fixed camera flip.

//for loop was set to 4 instead of cams[] length.

aug 15th 2017; tried to implement shipload from script. hit multiple bugs. probably best to retry from scratch...

//try to load by btn... if it works, its a timing issue.

aug 16th 2017; definetely a timing issue.

coult'n find good spot to call SetAllShips () function, but at least made it place all 8 ships from different tiles simultaneously...

//tried the btn. worked... but why?

aug 17th 2017; couldn't figure it out...

aug 18th 2017; FIXED IT!

//PlaceAllShips () was executed before Start () which defined the length of the ID[]. i repositioned that declaration directly to the Placement () functions. now it works, though i dont know why Start () isn't executed earlier...

aug 19th 2017; creating a networked lobby & select layout/ship

~~-create network manager in lobby~~

aug 20th 2017; assigning ID && fixes

~~-assign unique networkID to every player.~~

~~-get localPlayerID in NetManager~~

~~-fix Ship Load from Standalone //using consistant datapath & generated new problem -> savedata folder.~~

~~-add layout information to .sav file //i think~~

aug 21st 2017;

~~-change scene collectively~~

~~-dis-/enable ships based on numOfPlayers~~

~~-fix camswitch according to disabled ships~~

~~-fix bug that arose from dis-/enabeling~~

//the bug was caused by PlaceAllShips () since it tried to access all instead of only the active ships...

//testing shipLoad from remote machine.

-> emulate remote machine by using a version of the game that draws from resources (the unity editor ver) and one that doesn't

aug 22th 2017; Bug Fixing

~~//the layout loading bug appears only in dis-/enabled builds~~

~~-fix layout loading in standalone~~

//actually it had nothing to do with the dis-/enableing

-> it was the shipType string in the cashe script which was reassigned to null form the inspector, thus telling the system to load a ship called "". this issue only appeared when the cashe object existed, which it only does when the application is started from the main menu. thats why in unity the problem didnt appear but only in standalone apps.

~~-fix shipplacement according to numOfPlayers~~

~~-set start cams according to playerId~~

aug 23rd 2017; testing for bugs && recording thoughts

//documented on whiteboard [insert wB-ID]

aug 24th 2017; outlining remote ship loading

aug 25th 2017; implementing remote ship loading?

~~-give NetManager a counter int~~

~~-flip cams according to localPlayer //done? (need 2B tested)~~

~~-automate listing players and getting ID~~

//might want to check for bug later on... //moar tomorrow

aug 26th 2017; trying to implement [syncVar] to int [][]
//the netManager is no networkBehaviour. but a networkBehaviour
can be tricky when its a singleton. so i split it in two, the
manager <singleton>, and the behaviour, for syncing certain
variables. the manager has a reference to it. (lies on same obj)
//mayhap should just use playerInfo since its a networkBehaviour.
only problem could be that theres multiple instances in every
instance of the game...
//syncvar doesnt work with arrays, should use syncList instead
//okay, syncvar/list repeatedly updates, whilst i only need to
sync the information once for initialization. so ill use a
clientRPC what ever that is...
//RemoteProcedureCall. -> add entry to notebook... (later)
//not the best way to proceed.
//will use string to encode the int [,] and decode locally.
//the supposed maximum length of a string is 2^31 or 2147483648

//will try another method. back to rpcs. trying to build a
prototype in a new project.

aug 27th 2017; implementing syncing of loadedShips (hopefully)
//ini error!
//done a bunch o stuff, few bugs:
-the clients shipString array isnt properly set up
-always loads same ship (it seems?)
-cant list [,] in anyway. found a workaround, but it aint pretty
~~-implement counter functionality~~

aug 28th 2017; placement seems to work now... for no reason...
//only one problem is, it seems to be placing both ships on
player 2... will test with 3 players.
//confirmed
//pretty sure i found the source of the bug. i forgot to remove a
line of code placing ship0 on every layout...
//Player.Instance.PlacementLoop (ship0 , i);
~~-load non local ships~~
//it friggin works!!! (it seems, lets not jump to conclusions)

aug 29th 2017;

- ship selection //once ship is selected load layout
- //gonna settle with input string for now. -> needs to be changed to an array of btns assigned and named by .sav files in directory
- //order of operations:
 - add inputfield to lobby.
 - assign inputField the localPlayer.
 - give localPlayer a GetName () //should be getLayout
 - load layout string
 - sync layout string across network
 - feed syncedLayoutString to all instances of netManager?
 - nah, loop through players for this...
- //in layout placement loop:
 - //Place (netManager.Instance.playerList [i].layoutName);

aug 30th 2017 / aug 31st 2017; two day vacation.

//fuck you, i deserve that!

aug 31st 2017;

- ~~-add inputfield to lobby.~~
- ~~-assign inputField the localPlayer.~~
- ~~-integrate player instance in lobby~~
- ~~-give localPlayer a GetName () //should be getLayout~~
- ~~-load layout string~~
- ~~-sync layout string across network~~
- ~~-implement shipType override //for debug reasons...~~
- //i simply change the standard ship to load from gurren to concorde, since all ships can be placed there, then i use the inspector to edit the loaded layout to the proper one. ill do this for all 8 ships...
- ~~-fix RpcCall from client -> layoutName sync~~
- //LoadShipType () calls cmd that calls rpc. //easy
- BUG: meNoShipMassa: has problems loading shipLayout
- //-> the object calling the Cmd had no authority
- //-> because the playerId isn't set up at this stage, thus its always player0. need to find workaround...

sep 1st 2017;

~~//ship5 layoutString is fucked~~ //fixed biatch

sep 3rd 2017; //yeah i know, from now on daily

//not...

sep 5th 2017;

~~assotiate ID with layout~~

~~assotiate ID with ship~~

//it is by default... so yeah...

~~renew ship templates, so that they save the layout name.~~

//just re-save with the cashe script in-scene

//didnt do it today, but i forgot when i did so... looks productive though doesnt it?

~~implement ship selection~~

~~ship selection~~ //once ship is selected load layout

//if no ship is selected, readyState is fixed to false.

~~layout selection -> within save file...~~

~~fix ini issue, so shipName is properly assigned~~

//reassess ID every time a player enters or leaves...

~~sync with late bloomers.~~

//if a player joins after ships have been selected, send him a memo or something... otherwise bugTime!

~~//declare an List <string>... you know, thats stupid...~~

//if a new player joins, CmdIsThereSomethingINeedToKnow? ()

-> server loops through playerList.shipType and syncs it.

//how do i send strings to a specific client?

//CONSIDER IT DONE! //okay still needs to be cleaned up n stuff but eyyy...

sep 6th 2017;

//proably time to renew currency and placement...

-> added to TaskList

//outlined a system for placement synchronization

sep 7th 2017; fixed a few null reference errors in the hangar & added the missing four ships to the selection screen

sep 8th 2017; nah... not feelin it dude.

//need targeting sprite, a gun obj, a missile/laser obj.

-> for UI, and animation. although targeting sprite is the most important and should be displayed on all clients for debug puposes.

sep 11th 2017; synced targeting

//created target sprite, added type 5, and synced placement info

~~implement targeting~~

sep 12th 2017;

~~implement room hp, and damaging~~

~~implement system hp, and damaging~~

~~implement crew hp, and damaging~~

//add healthPoint script to all damageable objects, reference these objs by child.

-> for (all childs) { if (damageable) { damage(21) } }

-> damage (int dmg) { health -= dmg }

-sync dmg/hp

~~//mayhap its best if its only synced if another player is entering the room, or has scanners of high enough quality to improve efficiency // <- u stupid?~~

//damage has to be synced anyways? the owner of the ship needs to be informed of his ships hp. since he will rarely be the cause of dmg it needs to be synced anyways. the dmg value is sent to all players. the hp is calculated by each individually.

//the amount of dmg is currently hardcoded to PlaceTarget ()

~~-add healthScript to rooms, systems and crew~~

//the system assumes its no hit until it gets a confirmation from the target.

//order of operations: weapon is powered, target is set -> sync, weapon is fired unless retargeted -> sync or target is removed

-> sync. //the server determines if the target was hit. once it confirms -> sync.

//alt order: the problem with the server authority approach is, that last second input by the targeted player may be ignored. on the other hand it is probably a more efficient and maybe even secure option...

-remove destroyed objs properly

-implement multi-tile-room damaging

-sync Doorstate

-balance hp

sep 13th 2017; proper multi-tile-room damaging. //also birthday!
//so, heres the thing... only one tile contains the room obj no matter the room size. each tile contains information of the origin point though. to damage and properly remove the room, Damage () must be called on the obj. i use the localRoom info, to find the right tile, and access the first child, since rooms are always created first, making them "firstborn".
-> just keep that in mind should weird bugs occur kay?

//btw that brings up a new design decision: should weapons affect the entire room, only the tile and room structure or tile with diminishing damage output as it spreads across the room?
//the room would take sum dmg in the last one.

//new rooms? basically classic tile based sprites, so that each tile that technically already contains a room, also has a child obj that is part of the room. basically that would mean that every obj is 1 by 1 tile big, and only in collective form an image...

~~implement multi tile room damaging~~
~~//if empty tile is targeted -> null ref loop~~
~~fix nullRefLoop~~

sep 14th 2017; proper re-movement, //and synced doorstates; nope
//proper re-movement requires the obj type. ill store it in the health script. thats probable the easiest way.
//or i give each gameObj a specific script, each containing a removal function -> later on they'll probably need one anyways...

~~add room, system & crew script~~
~~remove destroyed objs properly~~
//i == idiot; i was so determined to properly remove rooms that i didn't realize it doesn't make any sense... instead of removing the obj, it should change its sprite to a more damaged version.

~~create damage sprites~~
~~add proper dmg visualization~~

sep 15th 2017; Doors and other calamities

//paths can be saved on the CrewScript, since no crew member has more than one path at a time.

//btw: doors always lie on the rooms origin.

~~-create a networked lobby //didn't really do that today but eh...~~

~~-add obj slots on tiles~~

//4 empty objs, room, system, crew, danger so that the obj type can be determined by the child index

//0 == room; 1 == system; 2 == crew; 3 == danger; 4 == door;

//needs a lot of null ref checks though...

~~-update removal functions~~

~~-fix empty removal exception~~

~~-make DoorState playerOnly //if (doorPos.Z == localPlayerID);~~

~~-give doors a placement function~~

~~-give doors a removal function~~

//why doesn't the door lie on the tween itself? should i fix?

~~-sync Doorstate~~

//would pathfinding be more efficient if it would scan the environment once, updating it should it change? -> probabs not...

sep 16th 2017; crew action selection

~~-create Sprite: crewSelected~~

~~-implement simple selection -> add CrewSelect script~~

~~-implement additive selection -> shift~~

~~-implement dragBox selection~~

~~-create dragBox Sprite~~

~~-implement dragBox function~~

//okay, so heres the deal... the dragBox checks only for the position of the objs. since the pos of the crew is bottom left its super counterintuitive...

~~-fix dragBox, objPos shit~~

~~-say dragBox aloud three times //to be continued... probably~~

sep 17th 2017; crew action selection 2

~~-fix dragBox, objPos shit~~

//second obj attached to crew, placed at center, given a 2d box

~~-implement dragBox selection~~

~~-implement selection into BattleVoid~~

//although for some reason it doesnt do simple clicks.. only drag

BUG: drag of the block: the dragHandler img blocks all placement.

~~-fix drag of the block //BUG~~

//simply disable img whenever clicked btn != null

~~-make selection localPlayer exclusive~~

BUG: input my face: hover and dragBox stopped working.

~~-fix input my face //BUG~~

//canvases cant be children of obj it seems...

BUG: noClick: cant select crew by clicking only by dragging...

~~-fix noClick //BUG~~

BUG: transDrag: if the image is mirrored, so is the dragBox

~~-fix transDrag //BUG~~

//btw only the main camera can currently display the dragBox

//would a better way of inverting the screen be to place the cam at -1 and rotate it by 180?

-> then the dragBox wouldn't be affected, but the drawing order would be.

//can i change the maths based on the cam?

//what happens if i change the canvas from screenSpace to overlay

->huh... that fixed everything actually...

//including transDrag //BUG

~~-implement crew selection~~

sep 18th 2017; CREW MOVEMENT

-implement Crew Movement

```
-fix A-star //~...~
    -give each crewScript path
    -implement multi-path finding -> if multiple paths are
      requested simultaneously.
      //probably best to call GetPath () from CrewScript
      -> GetPath (Point _crewPos," _targetPos, CrewScript _c)
      -> if (multipleCrewSelected || roomOccupied)
        { foreach { AssignClosestFreeTarget () } }
    -implement friend/foe recognition
      //-> for pathfinding && friends
    -implement movement animation
    //btw, when crew pops node, reassign it to that room.
    -> passingCrew obj.
    //there should actually be 3 crew obj: friend; passing; foe;

    ->if (Input.GetKey(Mouse (0)) && selectedCrew != null)
    { AssignTargetPos (); GetPath (); }

    -Update Obj References
```

sep 19th 2017; 2nd Movement

```
//if (goal != free) { CheckForClosestFreeX (Point goal); }
-> for now at least
```

BUG: HyperactiveShipMessenger: pathfinder calls PlaceAllShips ()

~~-fix HyperactiveShipMessenger~~

```
//unity activates the last clicked btn if (Input.Space) in my
case that was PlaceAllShips ();
```

-make crewPos nodes [start]

~~-Implement targetPos selection //when crew is selected~~

```
//if (crewSelected && Input.mouse (1)) -> setGoal
```

```
-> make that if (currentlySelected != null && mouseInput)
```

```
//actually the click check should be done by the tile... ->whiteB
```

sep 20th 2017; 3rd Movement

order of operations:

- player selects crew
- player selects destination
- all selected check
- //if (destination.IsOccupied)
- > by either crew or another destination
 - calc path
 - move

-implement gameState: hangar; battleVoid

//workaround: could i disable on Down and select on Up?

BUG: mr McMouse: the upDown workaround works perfectly except when using the macs trackpad rightMouseTap thingy

//edit: worked with pad once!

-fix mrMcMouse //BUG

~~-create bug register~~

~~-remove destination on reassignment~~

~~-getPath~~

~~-deselect crew upon death~~

~~-remove destination upon death~~

~~-give each crewScript path~~

~~-implement multi-path finding //when multipath~~

~~-select start & goal~~

//okay, few problems... nothing to worry about:

-the former destination is deleted, before its certain the other is reachable.

-the path gets returned even though i dont know if there's a way

//mayhap generate path before actually placing the uiObj.

-> should be easy enough to modify. its cool for now

//if destination is within the same room, then what?

BUG: PimpWalk: crew moves super weirdly!

-fix PimpWalk

//yeah sure fuckid! future self: up yours! you do it! why do i alw

sep 21st 2017; 4th Movement

//path log's clear. everything is the way it should be...

-> it's the animation part

//if path is changed { setPath }

//okay, so he knows he hasn't reached the goal yet, but doesn't proceed for some fuckin reason... yet sometimes he does?

//it stops although there is another point available to it

-> fuck a doodle doo

sep 22nd 2017; 5th Movement

//nada

sep 23rd 2017; 6th Movement -> coroutine

//could it be z?!

MOVEMENT FIXED!

//distance needed some wiggleroom! -> couldn't be (pos == dest)

//still some super weird bugs tough.

BUG: DanceTeddy: if destination is changed mid transit, weird shit happens...

~~-fix DanceTeddy~~

~~-include goal in path~~

//could alternatively add goal to stack twice -> will do

~~-unman rooms upon departure & man rooms upon arrival~~

BUG: YouAintMySupervisor!: crew increases speed when destination is reset mid transit...

~~-fix YouAintMySupervisor! //BUG~~

~~-remove destination UI tile upon arrival~~

//should crew have a networkID? -> would ease the movement sync

//it now has

~~-sync general movement~~

~~-implement PointToVector () && VectorToPoint ()~~

//although it seems to sync, there are still some complications with the movement speed (the remote players crew moves twice as fast) and the NetworkHash128 key in the crewDict (exists twice or not at all most of the time)

sep 24th 2017; collecting ideas & voting our new gouvernement

~~-create Idea compilation for Rogue Elements~~

~~-vote~~

~~-discuss design n stuff~~

sep 25th 2017; Last Movement //for now //hopefully -> nope

killing tuesday:

-kill tuesday

-fiesta!

BUG: ButtTargetMy: target positions dont remove properly upon arrival

~~-fix ButtTargetMy BUG~~

New System for Crew Referencing:

-based on the playerId crew adds itself to crewList, (on all Instances individually.) the movement order simply carries the index in form of an int

//-> since the placement order should always be the same.

~~-add crewList crewDict <int, CrewScript> to playerInfo~~

~~//make that dict nope!~~

~~-get crewList index to crew~~

~~//for (i = 0; i < crewList; i++) { crewList [i].index = i; }~~

~~-add crewList to playerInfo~~

~~-find out where to call GetCrewIndex ()~~

~~//is called when crew gets added. not the most elegant solution..~~

~~-fix no path found issue~~

~~-sync general movement~~

~~//clean up~~

~~-reassign ancestry crewObj~~

~~-create adaptive and reactive movement~~

~~//if room/system/elevator is destroyed, recalculate path.~~

BUG: NeverGiveUp: crew stops all movement when goal is unreachable. //they should move to their original target or go to the closest room.

~~-fix NeverGiveUp //BUG~~

~~-remove destination tile when unreachable~~

sep 27th 2017; Moar Movement

~~-fix unreachability issues -> fix NeverGiveUp //BUG~~
//dont interrupt current path if the destination is unreachable
~~-implement door struggle //waitForSeconds edition~~
//crew fights doors if closed -> give nodes doorstate, reevaluate when reaching node.
~~-> if (!localCrew || noDoorPower || !localShip){ DoorPamPam (); }~~
//not really
~~-improve crewPos transition.~~
-> when reaching tween, set crewPos to next node
~~-improve AStars bool acquisition~~
//give nodes the booleans of their tileRef

sep 29th 2017; Doors

~~-implement doDoorThing bool //for Debug purposes~~
-> can he open doors?
~~-open doors for crew~~

sep 30th 2017; Moar Doar

~~-add start twice for doorcheck~~
//works! //not that that surprises me in any way...
-rework rooms & placement
//so that a 3by1 room consists of three parts, each situated on its tile retrospectively, so that each part can be modified individually.
 ~~-come up with new refSystem~~
 -introduce new obj referencing system
 -recrop room sprites
 -redo roomPrefabs
 -update placement functions
 //-> new refSystem
-introduce reactor
-repair shit

oct 4th 2017; Placement Do-over

//actually just mapped some functionality

oct 5th 2017; Some Actual Work...

- rework placement reactions
- add objScript to all placables
- introduce "subsystems" //new name is in order

//-> everything that goes into system slots. //does that include elevators?

Order of Operations:

- get input //what to place, where
- check for conditions //placable? walkable? manned?
- place obj & save reference //instantiate, and create a reference
- ref.Ini (); //call the ini function on the obj

//-> create simple obj script, so that the script reference mustn't change...

- set bools of tile
- if objWidth > 1: place next obj //the obj itself has the reference for the next obj that continues the whole.

oct 7th 2017; the new order

//do one obj at a time. starting with rooms. -> giving them the information and anything else needed to make changes...

//use inheritance for obj/objClass/item

oct 9th 2017; ReRef

//tested referencing system using IPlacable interface -> works!

~~-implement bool CanHaveDoor~~

//if !CanHaveDoor && roomIsPlacedNextTo -> place wall;

~~-create tileComponent versions of Rooms~~

~~-create cross-placable references //nonSpecific Script~~

~~-reintroduce DoorPlacement/Removal~~

//a few other things: i recreated most of the placement functionality within the placable itself, so that the TileScript can relax every once in a while. a bonus is, that it is now possible to create non-rectangular rooms and systems with great ease, decide from within the inspector if a door can be placed on the component or not and in which order the components are placed and where...

//crew and other placables are next, followed by some clean-up

oct 10th 2017; Adding on ReRef

- ~~-add multi-tile (adaptive) systems~~
- ~~-add sub-system slots~~
- implement new placement in crew
- ~~-implement new placement in systems~~
- re-implement room obstruction check
- ~~-give originObj objDimensions~~
- ~~-create space check based on dimensionString~~

//for removal and placement.

- ~~-restructure placement to selfdescribing~~ //so that the vector2 describes the objs own pos relative to the previous one instead of the previous one dictating it to the next.
- implement proper removal <- what did i even mean by that?

//quick informal summary:
multi-tile systems have been implemented, and check for space.
systems and rooms still need a proper obstruction check, when removing. systems have been fitted with subSystemSlots activated by bool. still lack functionality though du to lack of actual subsystems as of yet...

oct 11 2017; persistency reloaded vol1

to do today:

- ~~-redo room removal check~~
- ~~-do system removal check~~ //still needs conditions though
- ~~-implement subSystems~~

//incl. placement removal and new obstruction rules

- ~~-redo crew~~
- redo elevators
- redo dangers
- redo ui
- redo persistency
- ~~-fix path finding again~~

//twens werent set to walkable. is new standard for twens

//summary: i've completely overhauled the persistency system.
Rooms and Systems are now saved by the following:
xy_Placable1, type, X, Y; ->xy define the dimensions, the name gets the object reference, X&Y hold the position.
old ships cant be loaded anymore. the system now serializes strings instead of int[,]. crew, subSystems will follow.

oct 12th 2017; persistency reloaded vol2

- > crew, subsystem, accessor persistency updated
- > everything now utilizes PlaceObj (_objStr); to place an obj //as opposed to PlaceRoom (); PlaceSystem (); etc
- back and clean up
- do the same with removal
- ~~-fix multicrew pathfinding roomSpaceCheck~~
- //if destination is blocked, crew now checks if theres an available destination in the same gridPos.Y... if not it checks the rest of the room. //it outsources the check to roomScript
- order tileChilds to: room; sys; subsys; crew; door; danger; ui; //change the scriptshit
- ~~-fix removal conditions~~

oct 13th 2017; Re-moval

- //-clean childs //make door child of room //DONT// -> didnt
- ~~-renew removal~~
- ~~-do dangers~~
- ~~-do targeting~~
- do damaging
- > btw: dangers and targets are not persistable //cant be saved
- gave room trueOriginPos -> tile that was clicked to place room //shit. current targeting is flawed. (probably)
- > cant do any child references. everything needs to be adressed directly.
- kill all humans
- ~~-do somethin bout the layering~~

oct 15th 2017; System systems

- ~~-create distinguishable system sprites //tmp~~
- ~~-create shield sprite~~
- implement system functionality
 - reactor
 - ~~-implement reactor capacity~~
 - shield

oct 16th 2017; power management 1

```
//int reactorUsage -> if (reactorUsage + x <= reactorCapacity)
{ reactorUsage += x; return true; } else { return false; }
-add weapons
    -implement weapon slots
        -create weaponSlotObj //-> will be is tileType
        -resync targeting
        -create weaponSprites
            -small //2
            -medium //4
            -large //6
            -epic //66
-add shield functionality
-add shield power management
```

oct 17th 2017; sprites

```
-create weaponSprites
    -medium //4
    -large //6
```

oct 19th 2017;

```
//found out why the projectiles where moving so laggy.
-> Debug.Log in every iteration of every obj. -> DUH
-fix projectileMovement
-find out how to properly scale and position the GunBtn
```

oct 20th 2017;

```
-find out how to properly scale and position the GunBtn
//now truly is a btn
-fix btn scaling
-implement actual targeting
-implement radial bombardement
//coefficient = screen.width //speed = distance*coefficient/time
-sync radialBombardement angle
-center targeting
-make shit work
```

oct 21st 2017; list-a-ringding

- sync attack angle
- ~~-center target~~
- destroy old target ui when taking aim
- damage rooms, systems, subsystems, crew
- implement power usage for weapons
- implement fire functionality //spreading/damaging/extinguishing
- implement shield functionality
- implement teleporter
- fix systemPlacement space check
- ~~-privatize gunAccess~~ //only the owner of the ship the gun is attached to should be able to fire it. duh...
- fix speed and distance for differing resolutions
- ~~-fix retargeting~~

//when retargeting all shots still in the notAir will disappear. their target has been destroyed, thus they dont know where to go

- ~~-get taskList up to date~~

oct 24th 2017;

- stuff i presume

//BUG target removal issue 0: the target is currently removed by the projectile, but what if its never fired? //fixed

- ~~-find out where to destroy target obj.~~
- ~~//if (!shotsFiredAtThisLocation) { destroy targetObj; }~~
- ~~-> find out how to message obj if shot was fired.~~
- ~~//bool wasFiredUpon: set by projectile itself.~~
- ~~//also add a counter, determined by weapon for multiprojectile target stuff. if (lastShot) { destroy(Target); }~~
- ~~}-> or: each projectile owns its copy of the location, the first one to arrive destroys it.~~

//simply save the position on projectile from Setup

- implement keyboardInput weapon referencing
- implement beams

//2 points determined by Screen res etc.

- > this needs to be 100% nsync
- implement teleBombs
- read a book just once in your life

oct 26th 2017;

~~-fix speed and distance for differing resolutions~~

//taking the gridField.sprite.x * gridField.scale.x seems to work

-> still, the projectiles arrive at different times.

-implement power usage for weapons

oct 30th 2017;

//solved GunLogic

-> isCharged { set { syncIsCharged (isCharged); } }

-> if (isCharged) { if (hasTarget) { Fire (); } }

//fire the changes is charged back to false

//when target is set -> if (isCharged) { Fire (); }

//weird bugs occur. -> complete overhaul of GunLogic necessary?

oct 31st 2017;

~~-sync weapon powerstates //needs bool, and location~~

-merge syncPowerstate and doorState? //-> also bool, and location

-sync weapon charge //bool also?

//the coroutine is called local-only, then updating the bool?

-> itd be impossible for other players to tell the chargeState...

//should the routine be located on the weapon itself? so that it

can be called by syncFunction? or, given an entirely new routine.

//needs the gunPos anyways, so can take the chargetime from obj.

-make chargeState visible to other players //someday

nov 1st 2017;

~~-explore colliders and coroutines~~

//basically i just wanted to fill this page...

//started work on the sleepless special, encountering old

problems in a new disguise. finally overcame my fear of colliders and actually came to appreciate them.

-> learned how to StopCoroutines and attach ui elements to objs.

nov 3rd 2017;

- ~~-implement charge loop~~
- ~~-sync loop~~ //seems synced... needs further testing though.
- ~~-implement preChargeTargeting~~
- ~~-syncChargeState~~
- //tryShoot() whenever the target is set or the weapon was charged
- ~~-display chargestate~~ //someday
- //try 2 img, one child of the other, then adjust the width of the child according to the percentage charged
- //shoot should be called in a synced manner?
- ~~-implement shield~~
 - ~~-spawn it~~
 - ~~-scale it~~
 - attach and configure collider //weird shit
 - create shieldScr2 //one for the obj, one for the system

nov 4th 2017;

- ~~-visualize chargeState~~
- ~~-visualize chargeState to other players~~ //? //->unconfirmed

nov 5th 2017;

- ~~-implement shieldHP and collision~~
- ~~-implement ChargeShieldCoroutine~~
- ~~-fix YouAintMySupervisor!~~ //BUG
- ~~-implement shieldPowerUsage~~
- ~~-give weaponSlot bool topGun~~

nov 6th 2017;

- ~~-implement powerManagement a la FTL~~
 - ~~-UI~~ //powerBars //where to place them, how to access them.
 - //once the ship was placed, do UI
 - ~~-firePower~~
 - ~~-shieldPower~~ //kinda//good chance that ill have to change it
 - ~~-sync shieldPower~~
 - ~~-power by UI~~
 - ~~-display Reactor Status~~

nov 8th 2017;

- implement shieldSys functionality
- implement damaging
 - visual
 - functional
- implement evading //once the projectile has reached its destination, the owner of the targeted ship calculates the chance, syncs the outcome and calls either the hit, or the miss function on the projectile. //from the sync
 - ~~-implement evasion chance~~
 - ~~-implement overshooting~~
 - sync evasionChance outcome //might be a problem
- implement engines
- implement cockpit

nov 9th 2017; //mightve found a solution to syncedProbability

nov 10th 2017;

BUG iJustDontCare: when redirecting power while shield is loading
//when power is reduced, if (currentPower == power) stopCoroutine
~~-fix iJustDontCare //BUG~~
-sync evasionChance outcome //cant use int for multiple projectile weapons. -> need different values for _prob
//use string instead?
//former tries of syncing yet to be properly removed

nov 12th 2017;

- implement engines //-> ship evasiveness
 - ~~-implement engine Powermanagement~~
 - ~~-implement custom ship evasiveness~~
 - ~~-sync engine PowerStates //i think...~~

//no matter how damaged the system, it always needs as much power as though it was fully operational...

nov 13th 2017;

~~-sync probability...~~ //could need some further work
//what if... i just generate int[512]
-> with all ints Random.Range (1, 100) and then sync that array
to all instances of that obj. whenever it shoots _counter++
//that way, all players have the same probability value without
having to constantly sync. //mayhap use stack instead
//if all values have been used, generate and sync a new one
or take the last used number and use it as indexer and so on...

//mayhap i should use the systems i put in place to achieve this
to sync whenever a target is set and generate a string of the
length of the maximum amount of shots fired at once to avoid
crazy desync errors

-revisit speed sync!

//something seems off. dunno why, but it just desynced

nov 27th 2017; it's been a while... SMASH!

-implement damaging, hp, and damage related powerBehaviour
 -kill all humans
 -implement crew interaction
 -implement crew teleportation
 -implement oxygen and
 ~~-reimplement objHP~~
 -implement capacityReduction
 ~~-reactor~~
 -weapon
 -shield
 -fixDeployedPowerIssue
 //if reactorCap = 6; weaponReq = 4; weaponPowered;
 reactorCap -= 4;
 -> weapon stays powered! once unpowered everything
 works fine again

dec 1st 2017;

-BUGS: systemOverload; randomMisReferences;

dec 4th 2017;

-BUG: if crew is moved outta way, theres a null reference on ammo
//-> crew removes first obj on ui layer when moved
-> store destination and target separately

~~-fix ImOutDontShoot! //BUG~~

-depower systems according to reactorUpdate

//SomeLogic: negative availablePower

one random powered system gets shut down;

if (availablePower < 0) { repeat; }

//whenever a system is powered, add to poweredSystems[];

whenever a system is depowered, remove from poweredSystems [];

~~//-> implement IPowerable interface //not yet necessary~~

-> use powerBtn instead

-BUG: if reactors are destroyed, it doesn't matter if its yours.

~~-fix whatsYoursIsMine //BUG~~

-BUG: displayed available power doesnt always add up

-BUG: if all reactors are destroyed, an error occurs.

//-> probs cuz no instance exists no mo'

dec 5th 2017;

-BUG: if the weapons are powered by the UI panelBtn, theres a null reference exception for the ammo...

-> the probability string isn't generated when powered this way

~~-fix IDontGenerateThere //BUG~~

-implement proper damaging //-> dmg -sprites and -functions

dec 7th 2017; proper damage

-dmg

-create dmg sprites for systems

~~-implement updateHealth() //in IPlacable? //aka damage();~~

//mayhap not damage but updateHealth? -> can also repair

//or have two different functions...

~~-unify health across entire room and all subRoomObj //?~~

dec 8th 2017; crew behaviour? <- nope

- ~~-implement placableRepairment //although no functionality~~
- implement roomHealthManagement
- implement systemHealthManagement
 - reactor
 - engines
 - weapons
- implement subSystemHealthManagement
 - elevator
 - ~~-nonFunctional~~
 - fix crewBehaviour
 - gooblebox (interface/terminal)
 - implement crewHealthManagement
- implement repairOrder //-> order in which placables are repaired
- ~~-determine ammoDmg by gun~~
- BUG: destroyed Elevators are a oneway ticket. others still recognize them. ->when downwards for no reason?
- fix elevatedToANewLevel //BUG
- BUG: crew needs to check path while following too. if elevator is destroyed after path was generated, crew will ignore.
//regenerate path whenever accessor is destroyed?
//check while walking
//what happens if elevator gets destroyed while crew is using it?
-> moves to closest elevator. regenerates path. if none is found it stays where it is. if there is no other elevator on this x, crew dies/gets stuck until elevator is repaired?
//actually, has it got to be a elevator? its in weightlessness anyways, would fix the carriage problem and wouldn't need power..

dec 9th 2017; elevatin'

- ~~-fix elevatedToANewLevel //BUG~~
- //-> node.HasElevator was't updated
- implement ISystem interface
 - ~~-shield~~
 - reactor
 - weapon
 - engine

dec 11 2017; dealin' some proper damage

```
-ISystem
  -reactor
  -engine
    -weapon <- might be tricky //there is no weaponSystem yet...
//should health be handled by origin exclusively?
//exchanged all 2s in engineScr for sysType //-> could cause t?
-implement Damage
  -engine
    -functionality
    -UI
  -weapon
    -functionality
    -UI
  -shield
    -functionality
    -UI
```

dec 13th 2017; healin' said damage

```
-implement repairs
  -crew
    -action
    -animation
  -engine
    -functionality
    //once 'twas repaired, it cant be damaged again //why?
    ->fix: isDamaged want updated when repairing...
    -UI
    //probably universally implementable
//implement ISysType? -> different damage stages for different
engine types for instance...
//can one by damaged systems, that need repair from the get go
but are cheaper //basically second hand or black market systems
//Devlog Note: some of it was done on dec 14 (the bool && UI)
```

dec 14th 2017; repair crew 00

//turn is manned into function?

```
IsManned () { isManned = !isManned; if (isManned) { repair(); } }
```

-> would work

//or put the function on the crew ->that way different crew can have different repair speeds etc

-implement crew behaviour

- implement repair loop in crew

//each crew has a loop, that just repeatedly updates one number by a certain amount. that way if multiple crew repair one system it adds instead of two bars being repaired separately. //where is that value???

//-> maybe small script: crewHandler on the crew subObj of the tile. //that way i just reference the parent for that. //NO! but instead, each placable does that on its own, that also means that a single tile obj like a terminal or accessor cant be repaired by multiple crew at a time. //the crewHandler could come in handy later tho.

- implement damage loop in crew

- redo UI //not by colours but by setActive and barScr

//-> each bar has a set amount of obj. each coloured. that way i can have the bars increase and decrease width according to dmgInProgress a la FTL

//IDEA: reddit inspired balancing and anti-Cheat

- implement auto move to terminal

dec 15th 2017; repair crew 01

- do tmpvisualRepresentation of system state. //1 fixed//1 dmgd

 - systems

 - rooms

 - elevators

 - goobleboxes

- ~~-fix FixShotKillz //BUG -> crew dies whatever happens~~

- ~~-fix FixOriginAlready! //BUG -> crew only repairs at origin~~

//actually crew only repaired departTile cuz o bad logic!

- ~~-implement repair loop in crew~~

- ~~-implement combinedEffort //multiCrew repairs~~

//SUMMARY: crew can now repair engines. big whoop!

dec 18th 2017;

- create crewRepair sprite //for crew //duh
- create crewFight sprite //just placeholders
- create crewRunning sprite
- ~~-fix MeToo //BUG //-> dmgd engine affects everyone...~~
- //fixed for now, maybe powerManagement should be decentralized and managed by the ships //perhaps i should test repair too...
- //-> that way other players can monitor your power management
- weapon keyboard shortcuts
 - ~~-power up~~
 - ~~-power down~~
- ~~-check if repairs work in mp //they do~~
- //btw: all the work put into reactor dmg is basically nillified
- ~~-outsource gunBtn logic to separate function~~

dec 19th 2017;

- ~~-limit repair to max hp //i think~~
- //->theoretically everything can be repaired to infinity aon...
- inform crew when all's repaired //if all's good, crew can chill
- //when something is damaged or repaired, it informs the room origin. the room origin then informs crew.
- //theres some weird bugs. need to fix dat... //tomorrow!
- ~~-inform roomOri about num of damages~~

dec 20th 2017;

- introduce isInoperable; //damaged doesn't mean it doesn't work..
- ~~-inform crew when all's repaired~~
- ~~-inform crew if repairs are needed~~
- ~~-implement repair loop in crew~~
- implement transporter //do soon reminder
- //to test friend/foe recognition, damageLoop etc.
- //on second though: its easier to test in one ship anyways...
- fix Hangar //shipCreation
- //im actually not sure if thats an easy task or not...
- //next: crew vs crew, crew vs interior, crew and terminals
- //also: crew movement tweaks and more systemTypes with fully functional damage system

dec 21st 2017;

~~-fix Hangar~~
~~-create second sample ship~~

apr 6th 2018; long time no see...

//**quick update:** it's been a while... no real reason. i thought i had one. and i probably did. but now i have gotten excited about RE all over again. im going to focus on the couch coop part of the game for now, simply because i think it might be more fun. the classic mode won't be cut though! it's just not my focus atm.

//i suspended classic crew movement. simply comment out UndoChanges () to 'unsuspend'

~~-implement mode selection in lobby //couch/classic~~
~~-spawn crew type dependent of game mode~~
~~-implement simple couch-crew~~
 ~~-movement~~
 ~~-terminal interaction~~
 ~~-aiming~~
 ~~-combat~~
 ~~-repair~~
 ~~-demolition~~
~~-make drag-box mode dependent~~
~~-reimplement classic movement /make movement mode dependent~~
~~-implement splitscreen~~
~~-implement controller support~~
//should i use more colliders? -> probably
//3 dimensional ships?! -> stowaways and harder ship navigation
~~-figure out how to spawn couch crew in networked system~~
~~-fix couch-crew scaling issue~~

apr 7th 2018;

~~-create couch lobby //room to bind the player to a screen portion~~
//where?! -> pre-lobby
~~-make main menu controller friendly //except it doesnt work for c~~
~~-introduce mode selection panel in main menu~~

apr 8th 2018;

- controller to player pos setup
 - create Dict <int playerID, string controllerID>
- ~~-"dynamic" split-screen~~ //a player joins, the screen is split
- //let me gather myself...
- a crew member is now assigned to a controller.
- >what if theres not enough crew members on the selected ship?
- ~~-Have casheScr save the playerID and controllerID~~
- //casheScr then listens for crew spawns and assigns the controls to them in spawn order
- ~~-assign controllers to crew~~
- implement couch friendly doors
- couch systems

apr 9th 2018;

- //added collider to fullspace
- ~~-block fullSpace~~
- ~~-block emptySpace~~ //had to privately rename Walkable to walkable
- ~~-implement couch friendly doors~~ //kinda
- make doors ask for keys //made door colliders trigger.
- ~~-bring back a star~~ //seems to work...
- implement elevators
- implement passable crew //player can walk past another

apr 10th 2018;

- ~~-create elevatorScr~~
- ~~-make elevators aware of one another~~

apr 11th 2018;

- implement IInteractable? -> elevators, terminals etc.
- //the interactibles will do the collision detection for now
- > no IInteractable interface
- ~~-create canvasManager~~ //for splitscreen UI
- ~~-create elevator menu~~ //sketch
- ~~-create couchCursor~~

apr 12th 2018;

~~-implement elevator menu into battlevoid~~
~~//elevator movement is fucked btw...~~

apr 13th 2018;

~~-fix elevator movement //its working, but ignoring the rb~~
~~-limit selectable levels //elevator panel~~
~~-fix multiCC issue //spawning more than one, fucks placement~~

apr 14th 2018;

//one year ago, the first ROGUE ELEMENTS folder was created.
//rogue elements is in production since feb 17th...

~~-implement couch aiming~~
~~-BUG: crewDestination wont disappear in damaged rooms... //fixed~~
~~//destination tile is weird when room took damage...~~
~~//repairing defect 201217 doesn't have this bug. -> couchMode~~
~~//it's IsStationed (); //repairLoop... //ill just redo repairs.~~
~~-implement localCouchDamage //when square is pressed, do damage.~~
~~-implement localCouchRepair~~
~~-tell them when to stop~~
~~//but doesn't have any effect yet... -> no dmg sprite and nothin'~~
~~-give rooms repair bar~~
~~-sync repair bar for all room modules~~
~~//all rooms share one repair bar //for now anyways...~~
~~-update couchCrewPos~~

apr 15th 2018;

~~-give damage an actual effect~~
~~-change roomSprites onDmg () //should redo that thing as well~~
~~//-> it just changes the origin spr to a broader one...~~
~~//-> outsource damage to hpScr?~~

apr 18th 2018;

- give systems, and subSystems hp bar
- implement new Dmg system as described in the physical DevLog
//-> basically sabotage damages all objects that lie on the tile crew is currently on. each "type" //rooms, systems, subSystems// can be called separately by crew.
- ~~-reroute sabotage over tile~~
- route hpBar over HealthScript

apr 19th 2018;

- ~~-route hpBar over HealthScript~~
- //give crew references to all originObj of the obj on this tile
- > know when to stop and such...

apr 20th 2018;

- ~~-give crew references to all originHScr~~
- ~~-give systems hp bar~~
- ~~-give subSys hp bar~~
- ~~-implement new Dmg system //except it doesn't do shit~~
- ~~-adapt sprites to damage //on all elements~~
- do proper subSysSlot check when placing systems
- //if you place an elevator first, the system may not be placed in a way that the elevator now lies on a system tile without a subSys slot...

apr 21st 2018;

- make elevators go kaputt
- overhaul PowerManagement //sure?
 - ISystem
 - proper damage
 - ~~-engines~~
 - shields
 - weapons
 - new powerManager //non singleton -> most def!
 - ISubSystem?
 - ~~-route gun damage the same as sabotage~~

apr 22nd 2018;

- overhaul powerManagement //<-yes, do that! its worth it!
 - //is it though...?
 - shields //shields now determine how many hp the shield has.
 - //for now, hp is in the hundreds and not as seen in FTL
 - //how to do shield alpha? //shields need recover timer
- //in cM systems are powered manually -> movement must be fun
- ~~-implement couchCrew powerUp //of systems~~
- //idea: have systems be turned on by literal switch? lectricChair
- implement damage UI
- BUG: when repaired, one powerBar is immediately powered...
- ~~-BUG: you can power down a damaged system... //fixed~~
- //ill fixit some other day...

apr 24th 2018;

- ~~-fixed it! //-> BUG: when repaired, system is immediately powered~~
- //damaging seems to work for now.
- reimplement shield
- reimplement weapons
- implement terminals as interactibles
 - couch aiming
- make couchMode mp capable
 - spawn and sync crew
 - //the system interactions can be synced through them...
- crew to crew combat
- have fixed ship size //take out adaptive scaling...
- //-> important for networked pos (?)
- implement reactor UI and such
- //something i forgot... but what? ah -> elevators
- do elevator interaction by crewPos instead of collider

apr 25th 2018;

-implement engines

-implement reactors

-proper reactor shutdown

//might prove tricky since power management is a one way road at the moment...

//fix? -> add all powered systems to a list located on shipPwrMgr, remove all destroyed or unpowered ones.

//need separate function or bool to power reactor

//the origin should know about the full system requirement?

//-> okay, i've made it worse...

-> back to the drawing-board...

//reactor fuel could cost shit, thus you might not want all to run full time...

apr 29th 2018;

~~-fix reactors and power management //i think i did anyways...~~

~~-fix damaging //doesn't power down properly~~

~~-fix reactor damaging~~

~~-implement proper system shutdown~~

may 1st 2018;

~~-implement engine functionality~~

~~-implement shield functionality //can't be hit though...~~

//well it can be hit, it just doesn't respond and guns don't work
//shields work similarly to the way they did in FTL. but the shields strength is determined by the system.

-> a system that requires 3 power could boost the shield by any amount really... //max shield is 8. any excess is bolster in case a system is destroyed. that may change though.

-implement weaponsSystem damaging

-implement weaponsSystem powerManagement

-implement weaponsSystem functionality

//ripped out a good chunk of the original weapon functionality...

-implement terminal functionality

may 3rd 2018;

- implement weaponsSystem damaging
 - ~~-systemPower~~
 - weaponPower //weapons don't automatically power down yet.
//or power at all for that matter... ^^
- implement weaponsSystem powerManagement
 - ~~-systemPower~~
 - weaponPower //weaponSys acts like reactor for weapons.

may 5th 2018;

- ~~-fix hangar shipLoading~~
- implement terminal functionality
//are terminals systemType specific or can they be applied to anything and their functionality is then determined by the system they've been place on?
 - weaponsTerminal
 - ~~-terminal interaction logic for couchCrew~~
 - ~~-camSwitch to fullShipCam~~ //currently fullscreen only
 - cursor for aiming
 - placing targets by controller
 - loading weapons
 - selecting weapons
 - powering weapons
 - charging weapons

may 6th 2018;

- weaponsTerminal
 - ~~-cursor for aiming~~
//cursor collision is detected by room
//couchTarget placement works differently from Classic
 - ~~-placing targets by controller~~
- //out-commented tileScr.GetGun
- //weaponSlot has and needs box collider for seemingly no reason.
->fix it!
- ~~-fix roomCol~~ //roomCol needs to be deactivated in hangar
- ~~-implement 3mode structure~~ //hangar, classic, couch

may 7th 2018;

//when terminal is used -> sys is powered.
//for weaponScr that means, that you may distribute any available
weaponPower //but cant aim without a terminal
//players can only do that by using a terminal.
//-> there may be weaponSys without terminals, but you can't
interface with it, just boost the available power
~~-firing weapons~~//though without random acc, with just one gun,
just one player, etc. **//but damaging works!**
-implement terminal kick //kick user from terminal if system or
terminal is destroyed or unpowered

may 10th 2018;

-implement couch mp weapon usage
~~-implement splitScreen shipScreen~~//spawn an instance of cam?
//relocate cam -> size and position copied from shipCam?
-> save standardCamValues
//no can do -> pos is relative to player -> 2 cams
//all couchPlayers have 2 cams, a crewCam and a ship cam
that becomes the child of the monitored ship -> spawned by c
~~-implement weapon selection~~
-weapon charge feels hinky //whenever the gun is changed, it
recharges, but is still fireable if the gun was previously
charged. when i tried to fix the issue, the guns stopped
working after a few shots fired...
//but switching weapons isn't supposed to power the weapons
in the first place, so it's just a not yet coded issue...
-reimplement online mp
~or~
-implement ai/enemy ships
-implement crew damaging
-implement crew to crew combat
-fix fullSpace elevator obstruction //elevators can move crew
through fullspace tiles. they shouldn't...
-get rid of ship scaling //have one size, thats it... let unitys
cameras handle the rest.
-create shipFrame creation software

may 11th 2018;

- uncollide couchCursor and weaponSlot
- fix weird inbetween collision issue
- //if the cursor is very between tiles, the collision won't work.
- implement weapon powerManagement
- ~~-stop charge coroutine properly~~
- //weapon charge behavior makes no damn sense. -> rewrite
- fix pathGoal assignment //in classic
- //dafuq!? the damn thing simply won't work properly...
- > same issue as i have with the couchCursor? collider issue?
- //could be some collider obstructing the tile...
- ~~-give active weapon colour coded outline //may be removed~~
- ~~-multiuser weapon management //if someone uses the weapon, it~~
can't be used by somebody else? or is can multiple people use the same weapon?
- //already targeted rooms should still be targetable
- //missed shots should also remove their target obj.

may 15th 2018;

- ~~-implement mp powerDisplay~~
- implement synced powerManagement
- spawn couchCrew
- implement couchTargeting ship switch
- //switching between ships in couchTargeting

may 16th 2018;

- ~~-fix resolution //no more var! just fixed ^^~~
- //alter cam to fit ship, instead of altering ship to fit cam...
- //make cam a child of displayField?
- spawn couchCrew
 - ~~-spawn hostCouchCrew~~
 - spawn clientCouchCrew
 - //client can't spawn!?
- fix shipBuildIssue //ship won't be placed if build is host...
- //-> if editor is client!? wtf?

may 17th 2018;

-spawn couchCrew

-spawn clientCouchCrew

//client can't spawn!? //-> it would seem that it isn't possible to send obj from clients to servers?

//it's the only explanation i could come up with...

//good news is, server side spawned obj work fine and are synced properly... they are controlled only by the local players etc...

//-> fix: spawn crew by sending a string to the server, the server instantiates the obj, passes the owners info to all clients and only the owner can runs CouchCrewSetup ();

//or place crew only on server, then distribute placement rights? -> no! cuz i need dem couchPlayerCounts...

//could sync those though... is it worth it though?

-> it might be since you might want to know the size of the team you're up against? but then, the way it looks now, you would anyways...

//seems to work... but the controller assignment is too fast for this spawning method... i'll have to send the string ` with it //place crew last?

~~-fix slowNetSync issue~~

//rigidbody sync is not the way to go it seems...

//crewPos isn't set!

//okay... that's it for today... it didn't work sadly.

although the cam was set properly this time, the clients position isn't synced, the servers position has an offset...

//i'll just set the networkTransform back from rb to pos

//client crew doesn't have authority!!

-> cuz it was instantiated on the server!

//can it be overridden? //how do i do that?

may 18th 2018;

~~-fix build to unity placement~~ //just extended artificialDelay ()

~~-fix client crewPos update~~ //if (isServer) { SyncPos (); }

~~-fix authority issue~~

-fix editor/build posDesync issue

-sync dmg, rpr loop //all crew actions for that matter

may 19th 2018;

- implement classic/couch compatibility //taking and syncing modes
//how to sync damage and power...
- sync dmg, rpr loop
 - ~~-sync damageStates~~ //isn't too helpful though, should be handled through health sync. //no states
 - > every other way would probably lead to bugs...
 - //all obj could sync every now and then, if its not fully damaged or repaired.
- //not by couchCrew though, but rather by GridPos //oldschool
- reimplement powerManagement in classic //just btns right?
- implement basic classicCrew behaviour
- //the players netID has reign over the powerSync
- //reroute system update to pass SysScr
- //should classicCrew also have a netID? -> syncing health and other shit... //doesn't necessarily need to sync transform though, right?
- //health updates: increase/decrease by amount.
- //but sync current hp every now and then...?
- //-create tile.hScrDict <int, HScr> //for easy health referencing
- > i did that already
- //022 room ain't workin'...
- visual power reference
- //so you can tell by looking wether a system is powered or not...

may 20th 2018;

- ~~-visualize powerState~~
- ~~-give crew healthBar~~
- ~~-sync hp loops~~ //dunno how reliable 'tis though
- //022 problem probably is, that there's no room segment on the tile that was clicked to place the room? //it definitely has something to do with pos, since it works if you stand on it's origin...
- ~~-fix problem022~~
- implement tragetedShip switch
- properly power weapons
- sync weapons //to stark time dilatation?
- ~~-fix height desync~~ //crew height is baaaad

may 21st 2018;

- ~~-implement targeted ship switch~~
 - ~~-cam switch~~ //no cam inversion though...
 - ~~-cursor thing~~ //so it moves with the cam.
 - //haha terminal controls targeted enemy weapons -> fixed
 - //enemy ship can be targeted, yet breaks weapons...
- //weapon time discrepancy needs to be addressed!
- //will create ship center obj for cursor
- //make everything server based //no client authority in damaging etc. all damage is synced by the server.
- properly power weapons //dis has become kind of a runnin gag ey?
- implement **couch** crew to crew combat
- implement teleporter
- ~~-reimplement shield collision~~
- //should probably be synced by server //might be a problem though
- //note to self: try to spawn full ship some time...

may 22nd 2018;

- //the cursor targeting does its thing, but not great...
- implement proper weapon power
 - ~~-limit power able weapons by available power~~
 - //d-pad for weapon switching //will do that shit tomorrow...
 - power weapons
 - unpower weapons

may 27th 2018;

- //yeah yeah, i was sick alright?
- ~~-implement proper weapon power~~ //unpowers twice. //sync takes too
 - ~~-weaponSys.dmg weapon powerdown~~ //when a weapon sys is powered down or damaged, it needs to check if there's still enough power available to power the powered weapons.
 - //neednt be synced! cuz wSys.power is synced! but not which!
- ~~-set up controller~~ //if random weapon is shutdown, sync is needed
 - setup d-pad //not possible in standard unity it seems...
 - ~~-setup shoulder buttons~~ //huge gridPos desync issue! -> fix!
- //funny bug: weapons can't unpower unless theres enough power to power the weapon up... easy fix though.

May 28th 2018;

- fix pos desync //crew seems left-er than is //editor issue?
- //spawn ships? //sync ship pos.
- ~~-fix client altitude issue //y isn't synced for clients?~~
- //-> crew pos is updated from server, where the elevatorCR isn't run. thus the pos isn't updated etc...
- fix speed issues //what's up with those?
- //resolution dependent. super weird.
- ~~-fix reactor tell //won't go grey no mo~~
- //ships wont place from build ...again
- //unity restart fixes the issue though...
- spawn ships //for positions sake...
- //maybe there's a workaround... just realign the ship based on the position of the tile [firstSpawnedCrew.crewPos] to firstSpawnedCrew.transform
- update unity
- sync projectile pos //or at least speeds!
- outsource crew controls //nah fuggid

may 29th 2018;

- //put couchCrewMovement in coRoutine? -> dont work in update
- //FUCK VISUAL STUDIO FOR REALS!!!**
- damage crew //by room collider //don't work, but why?!
- //works now, but doesn't //okay so, if i do the add on enter remove on exit approach, there's moments where crew is hurt twice from one projectile if it's in the room.
- //fix: half the dmg. if crew is about to exit, crew is less hurt?
- //except if the room is the direct hit.
- //this'll be more complicated im afraid...

jun 3rd 2018;

- implement fullRoomColliders //room to crew damage communication
- classic
 - classic dmg loops
 - ~~-reimplement power by button~~ //add systems to lists on mngr
 - //sysScr isPowered isn't working properly
 - fix raycasting //couldn't i just use the fullRoomCollider?

jun 4th 2018;

~~-classic~~

~~-classic dmg loop~~ //just based on arrival state though
//either crew needs to check regularly, or rooms inform
local crew about health state...

~~-implement friend/foe recognition for shipDmgLoops~~
//just a friend bool as of now...

~~-power weapon by btn~~ //doesn't yet work properly //fixed
//classic still utilizes the gunPOs approach. works fine,
but it'd be great if i could properly assign them their
weaponID...

//have btns created by ship after weapons where added to get
weapon id for targeting...?

jun 10th 2018;

~~-implement fullRoomColliders~~

~~-implement fullRoomCol~~ functionality

//why do crewOuch by collider tho? //still useful for beams
//annoyingly, the crew pos assignment goes out of bounds and
doesn't seem to be all that reliable...

//though it worked once! wtf!?

~~-fix raycasting~~

~~-new outlines~~ //why'd i do that instead of the important stuff?

//i don't fucking know!

~~-sync ship pos~~

~~-hurt crew~~

jun 11th 2018;

~~-fix crewPow reassignment~~ //cuz it's fuuuuck'd

~~-hurt crew~~ //crewDmg is not yet separated from roomDmg though

//let's sort thoughts: reassign crew only to room origin and only
when leaving or entering rooms //not tiles//-> more efficient?

//i'd need roomIDs back to make that work i guess. or just

compare origin objs... //-> let's do that, might not be the most
efficient way though. **//have the split-screen un-split on death?**

//UpdateCrewPos (int _x, int _y) is a thing now... ~~and z as well.~~

//nah, teleportation and shipSkip gets its own function.

jun 12th 2018;

- implement classicCrewOuch
- implement win state -> all crew ded

jun 14th 2018;

- ~~-fix raycasting~~ //although classic targeting sometimes defaults to room origin... don't yet know the cause of this...
//simply elevate tileCol? //doesn't work for like... no reason.
- ~~-fix room removal~~ //let's face it, haven't tested it, but... yeah
//just disable full room colliders? //easy fix, but maybe i should reorganize placement a little. but that's polishing to be honest. works for now...

jun 15th 2018;

- sync shipPos //i'll do this via netmng for conveniences sake
//-> truly spawn ships in future.

jun 16th 2018;

- //it's not shipPosDesync...
- //could the issue be that crew always is a child of spawnTile?
- //doesn't seem to be the case... i'll check crewPos in regular intervals to get some idea of what's goin' on..
- //clients couchCrew parents aren't set properly?
- //placeObj isn't called on clients
- call placeObj on client couchCrew
- //how? ->dunno
- //maybe stop assigning parents to couchCrew altogether?
- ~~-fix xPosDesync~~ //super ugly fix, but it seems to work...
- //fix janky movement. //varying speeds and shit. it's distracting

jun 23rd 2018;

- implement medBay
- implement teleporter
- ~~-implement classicCrewOuch~~ //and classic PosAssignment
- redo classic PosAssignment

jun 24th 2018;

-redo classic PosAssignment //by simply using the nodes?a <- duh?

jun 25th 2018;

~~-redo classic PosAssignment~~ /by nodes/ //whenever crewPos is set
//if _tile = tween -> extrapolate direction
//prevPos.x = 6; _tilePos.x = 7; => currentPos = 8;
//prevPos.x = 6; _tilePos.x = 5; => currentPos = 4;
//not super efficient -> performs double checks most of the time

//bar colouring is weird -> red-bars won't appear properly
//functionality is still fine though...
//weird bug: if reactor is destroyed by cannon, another powers up
//dunno what caused it... wasn't an issue this time... try build
//there's still posAssignment issues... sometimes crew isn't
properly removed or added... that is annoying as fuck.
-> redo move() coroutine... //at least partially...

jun 26th 2018;

//maybe debug classicPos by colouring occupied rooms?
~~-implement medbay~~
-implement teleporter
//couch crew hScr is never properly removed from spawn room
//-> might be added twice? -> fixed, it was added twice.
//that's what i just said fuckwit!
//who you callin a fuckwit pissface?
-BUG: DisRummSpecial //i forget. ah in hangar special rooms freak

jun 27th 2018;

//classicHScr assignment seems to work except elevators
~~-fix classicCrewElevatorPosAssignment~~ //-> there are no tweens!
//is there a way know crews mode of transportation? lets find out
//hScr is reassigned on elevator enter instead of exit //problem?
-syncCouchCrewPos and CouchHScrAssignment for elevators
-redo CouchUpdateCrewPos to take nextPos instead of increments
~~-fix weird origin fetish of pointerClick~~ //caused by fullRoomCol

jun 30th 2018;

- implement teleporter //crew thats just moved though the room is also transported. //including foes //?
- //design: teleporters have a carry capacity, that is determined by teleporter type, and power-level/condition. //should the teleporter be destroyed while porting, severe injury or death of crew follows. risk of injury or death increases with decrease of teleporter condition //hp wise. //!design//

jul 1st 2018;

- implement teleporter
 - ~~-implement teleporter system~~ //placable, powerable
 - implement teleport targeting
 - implement teleport functionality
- streamline implementation of new systems //?
- //currently two references in both battle-void and editor and a dict entry are needed...
- ~~-fix medbay power bug~~

jul 2nd 2018;

- implement teleporter
 - implement teleport targeting
 - classic
 - ~~-couch~~
 - ~~-terminal~~
 - implement teleport functionality
 - classic
 - to
 - from
 - couch
 - ~~-to~~
 - from
 - syncTeleportation
- //new strat: teleporter with external terminal, and airlock
- > simply teleport enemy crew in, and jettison them out.
- //counter strat? //anti teleport gear? //mag boots?

jul 4th 2018;

- implement couchTeleport functionality
 - ~~-to~~
 - ~~-from~~
 - crew dispersement //un-clump teleported crew
- //hand to hand: l1/r1 to loop through the hScr in current room.
- //skip friendlies?
- implement hand to hand combat
 - ~~-self harm~~

jul 5th 2018;

- implement hand to hand combat
 - ~~-harm of others~~
 - auto-unselect crew that left the room
 - //how though? give hScr a list of attackers, that are informed onLeave ()? //or check whenever anyone leaves the room? -> hScr has Check () called by room whenever the list is set.
 - visual representation of selection //targeting crew only?

jul 12th 2018;

- implement hand to hand combat
 - ~~-auto-unselect crew that left the room~~
 - //how? targetedCrew gets list of aggressors?
 - //if oneself is targeted, no-one is.
 - //solved, but dirtily... it's far from perfect and i have my doubts this is the way to go, at least concerning couchCrew.
 - //maybe have crew hit the target most, but if there's others nearby, they might get hurt as well, but less hard and based on chance?
 - visual representation of selection //targeting crew only?
- implement multi-ship in single-player //necessary for ai anyways
- look into PARSEC! //awesome stuff that might render my networking efforts useless...
- look into REWIRED! //controller software for unity. it is handled differently and will require some changes. the sooner, the better...

jul 14th 2018;

- implement classicTeleport
 - implement teleportBtns //in and out
- ~~-implement classicMedbay //? -> might already work //yup~~
- implement classicCloseCombat //utilizing readTheRoom () //get it
//give room bool containsFoe
//if (room.containsFoe) { if (!localTile.cF) { fList[random] } }
- implement ai
- implement "fun stuff"
 - fire that spreads
 - different weapon types
 - system variations
 - terminal minigames //? //terminal cancer... fuck... fuck.
 - oxygen!
 - sucking!!! //yay!!

jul 20th 2018;

//classicCrew auto use teleporter if (noPathFound)?

- implement fullCouchSupport!
 - selectable number of players //duh but kinda done right?
 - support 8 players //currently at 4
 - selectable number of ships
 - implement player to ship assignment
 - support multiple screens
 - //and yes... ai as well...

//issues: -all's geared towards networkedMP //->tons of rewriters
//ideas: -make repair/movement speeds dependent of crewNum?
//btw: why did i wait so long to implement this? -> some things
would've been so much easier to test actually.

//go check ship- && crewPlacement functions...

jul 25th 2018;

```
//spawn ships with localID and localCounter?  
//do ships have to be associated with localID? all that matters  
is if all crew in one team have died...  
-implement couchCrew team/ship assignment.  
//where is couchCrew spawned again?  
//crew has to be really fucking expensive.  
//for debug: color crew based on ship assignment.  
//maybe just have the one input field for ship selection, but  
once an input was made, it's saved to a list. it's also displayed  
and may be removed again. that's how the number of ships is  
determined...  
//im rerouting the load ship function to a new lobby script...  
  
//still need to add the ships to a list, remove them from that  
list on destroy and make those ships spawn properly. also crew  
assignment...
```

aug 4th 2018;

```
//i can explain. really!  
-multiship //spawn multiple ships from one instance of the game  
//might be best to load the entire shipString in lobby, send it  
to the server. then redistribute OnEnter(BattleVoid)  
//for now, it doesn't matter whose ship is whose...
```

aug 14th 2018; happy anniversary!

```
//tons of stuff! you wouldn't believe...  
//commented out NetManager.OnPlayerConnected () because of unity  
2018.3 ///? //they changed NetworkPlayer for some reason...  
//performed a quick check, everything else seems to work properly  
//OnPlayerConnected() only logged the ip. doesn't seem to be  
important...
```

aug 23rd 2018;

```
-update unity  
//hurray...
```


aug 29th 2018;

```
-implement multi-shipSpawn
//so...  -load the shipType && shipString in lobby
          //don't save shipString on playerInfo but cacheScr
          -> distributed through playerInfo though.
          //for polish: distribute shipType and name to clients
//infinite spawnable ships //just make it so you can spawn
ships and they automatically set up their ui, position, etc...
```

sep 3rd 2018;

```
//what if i were to spawn all objects?
//-> nonsense! spawn the ship, and count the objects placed,
assign them their count-number. //since all instances are placed
in the same order, the ID is valid.// then call through ship to
all clients. //-> full server authority. //i have no idea what
i'm talking about, but hey... at least it sounds important.
//does it make sense to do the multi-ship placement before
reworking the shipSpawn stuff? //it might.
```

sep 6th 2018;

```
//will do multi-ship now!!!
//i'll set up a shipClass //-> shipName, shipType, shipString
//client loads shipInfo, passes to server, server distributes by
PlaceShip (name, type, str, owner);
-multi-ship
  -implement shipInfo distributionSystem
  //maybe just spawn lobbyShipInfo?
  -clean-up //remove all *retired* placement/loadShip ()
    -fix bugs
      -powerBar issue
      -crew control
  -SpawnShips ()
  -Implement RpcPlaceShips ()
    -local
    -networked //is fuuucked... there's just tons of bugs.
//i will neglect the networked aspects of the game for the time
being... //what happens in classic?
```

sep 8th 2018;

- ~~-create network manager in lobby~~
- create a "readying matrix" <- that aint no matrix son...
- //if all players set themselves ready, the game starts.
- ~~-assign unique networkID to every player.~~
- //ID = int ranging from 0 to the amount of players - 1
- & ID needs to stay in existence
- get localPlayerID in NetManager
- implement ship selection
- add layout information to .sav files

-> tasklist:

- ~~-load ships from script~~
- ~~-fix shipLoad~~
- ~~-find way to time ship placement~~
- create a networked lobby
- assign IDs to Players
- load non-local ships
- animate crew
- finish this list