HEAVY STATION KIT

GUIDELINES
2021

CONTENT

3 Combination

5 Comparison

Prefabs:

6 General Information

6 Base Prefabs

11 Hangars Prefabs

16 Colony Prefabs

20 Dugout Prefabs

24 Dugout Blueprints

25 Blueprints Setup

Materials:

28 Base

29 Hangars

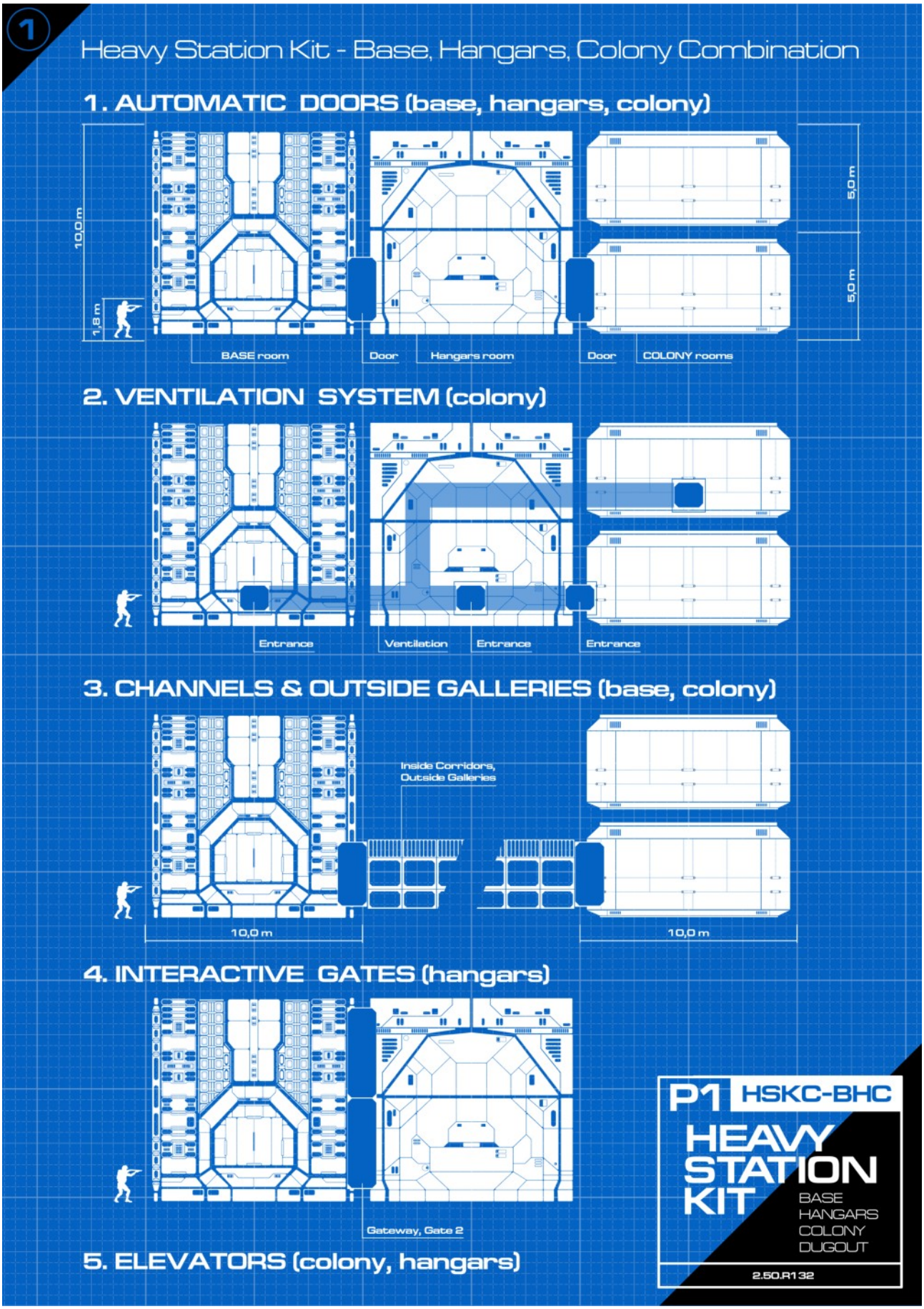
30 Colony

32 Dugout

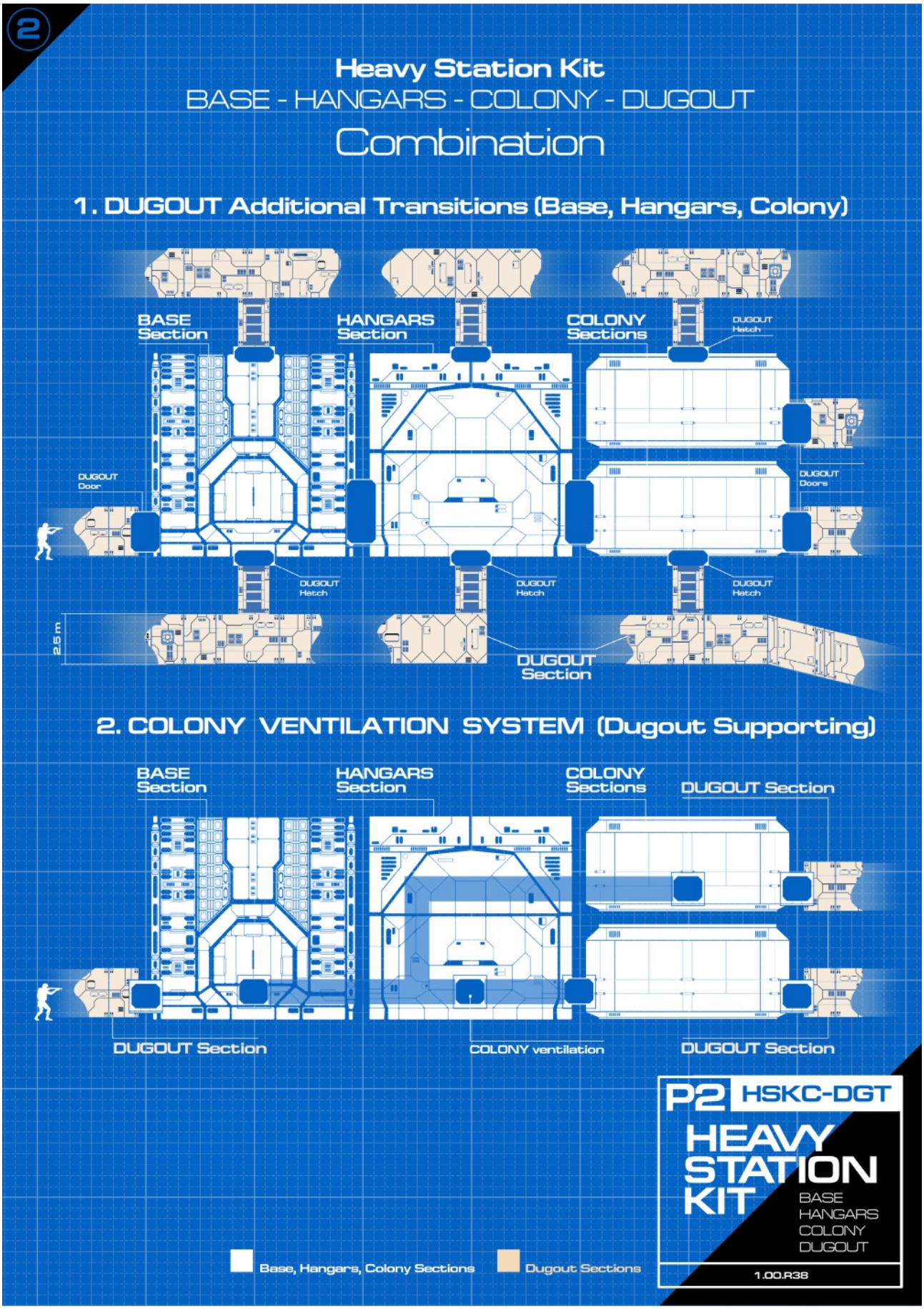
33 Scripts

44 General settings for 3rd party FPC

46 Installation Guidelines


1
Heavy Station Kit - Base, Hangars, Colony Combination
1. AUTOMATIC DOORS (base, hangars, colony)
10,0 m
1,8 m
5,0 m
5,0 m
BASE room
Door
Hangars room
Door
COLONY rooms
2. VENTILATION SYSTEM (colony)
Entrance
Ventilation
Entrance
Entrance
3. CHANNELS & OUTSIDE GALLERIES (base, colony)
Inside Corridors,
Outside Galleries
10,0 m
10,0 m
4. INTERACTIVE GATES (hangars)
Gateway, Gate 2
5. ELEVATORS (colony, hangars)
P1 HSKC-BHC
HEAVY STATION KIT
BASE
HANGARS
COLONY
DUGOUT
2.50.R132

Combination 2



Heavy Station Kit PACKAGES

COMPARISON

Availability of the unique Elements and Acceptance of the other Packages

CORE CONSTRUCTION

	BASE	HANGARS	COLONY	DUGOUT
Floors	Unique Eligible for Hangars & Colony Accepting Hangars & Colony	Unique Eligible for Base & Colony Accepting Base & Colony	Unique Eligible for Base & Hangars Accepting Base & Hangars	Unique
Walls	Unique Eligible for Hangars Accepting Hangars	Unique Eligible for Base Accepting Base	Unique	Unique
Arches	Unique Eligible for Hangars Accepting Hangars	Unique Eligible for Base Accepting Base	Unique	Unique
Outside walls (top-down theme)	Unique Eligible for Hangars & Colony Accepting Hangars & Colony	Unique Eligible for Base & Colony Accepting Base & Colony	Unique Eligible for Base & Hangars Accepting Base & Hangars	Vacant
Supports	Unique Eligible for Hangars Accepting Hangars	Unique Eligible for Base Accepting Base	Vacant	Unique
Partitions2	Unique Eligible for Hangars	Vacant Accepting Base	Vacant	Vacant
Perimeter Fence	Unique Eligible for Hangars, Colony & Dugout Accepting Hangars & Colony	Unique Eligible for Base, Colony & Dugout Accepting Base & Colony	Unique Eligible for Base, Hangars & Dugout Accepting Base & Hangars	Vacant Accepting Base, Hangars & Colony

TRANSITION FACILITIES

	BASE	HANGARS	COLONY	DUGOUT
Doors	Unique Eligible for Hangars & Colony Accepting Hangars, Colony & Dugout	Unique Eligible for Base & Colony Accepting Base, Colony & Dugout	Unique Eligible for Base & Hangars Accepting Base, Hangars & Dugout	Unique Eligible for Base, Hangars & Colony
Gates	Vacant Accepting Hangars	Unique Eligible for Base	Vacant	Vacant
Gateways	Vacant Accepting Hangars	Unique (10x10 and 20x10 meters) Eligible for Base	Vacant	Vacant
Ventilation	Vacant Accepting Colony	Vacant Accepting Colony	Unique Eligible for Base, Hangars & Dugout	Vacant Accepting Colony
Stairs	Unique (10 meters for Floor) Eligible for Hangars & Colony	Vacant (10 meters for Floor) Accepting Base	Unique (5 meters for Floor) Accepting Base	Unique (2.5 meters for Floor)
Ladders	Unique (10 meters for Floor) Eligible for Hangars & Colony Accepting Hangars, Colony & Dugout	Unique (Small Garage Ladder) Eligible for Base & Colony Accepting Base, Colony & Dugout	Unique (Swimming Pool Ladder) Eligible for Base & Hangars Accepting Base, Hangars & Dugout	Unique (2.5 meters for Floor) Eligible for Base, Hangars & Colony
Elevators	Vacant Accepting Hangars & Colony	Unique (10 meters for Floor) Eligible for Base & Colony Accepting Colony	Unique (5 meters for Floor) Eligible for Base & Hangars Accepting Hangars	Vacant
Channels	Unique Eligible for Hangars & Colony	Vacant Accepting Base	Vacant Accepting Base	Vacant
Galleries	Vacant Accepting Colony	Vacant Accepting Colony	Unique Eligible for Base & Hangars	Vacant
Ceiling & Floor Entrances	Vacant Accepting Dugout	Vacant Accepting Dugout	Vacant Accepting Dugout	Unique Eligible for Base, Hangars & Colony

Other Themes of the Prefabs
(Equipment, Furniture, Decorations, Objects, Props, etc.)
can be used in any of the Packages:
Base, Hangars, Colony and Dugout

PREFABS

General Information

The side and the height of the smallest cell or room possible is 10 metres (only the COLONY has 5m ceiling height).
If the scene is new, for just snapping the prefabs, we do recommend start building at the position x(0) – y(0) – z(0).

When you do this, most of the prefabs will appear right at their place. Some regular edits at the building of the cell:

- Walls and the like may be duplicated and rotated into the desired position
- The arches and doors will require 5m adjustment to the desired direction
- If the Top-Bottom prefab is placed at the ceiling, then it should have 10 metres offset by Y (5 meters for COLONY), 180 rotation by Z or X

At building the second cell there is 10 metres offset, because the side of the cell is 10 metres. So it is possible just duplicating the existing prefabs that are close to the position, and setting the required offset.

The Heavy Station Kit AUGMENTED Packages (Base+Hangars+Colony+Dugout) has **1672** Prefabs.

The Heavy Station Kit Packages (Base+Hangars+Colony+Dugout) has **1158** Prefabs.

The Heavy Station Kit duguot AUGMENTED comes with **131** Blueprints.

The Heavy Station Kit duguot comes with **110** Blueprints.

Heavy Station Kit BASE Prefabs

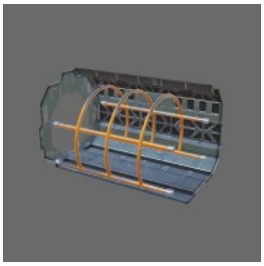
The Heavy Station Kit base 2.50 AUGMENTED has **318** Prefabs:

The Heavy Station Kit base 2.50 has **195** Prefabs:

ARCHES

Prefabs	Tris (LOD 0)	Colors	Notes	Position
18 7	450 – 2316	5	*Customize the color of the vertical elements.*	Position X 5 y 0 z 5 Offset X 10 Y 10 Z 10

There are intentional gaps between the walls. Arches do fill these. Also they may work as visual strengthening of the level.



CHANNELS

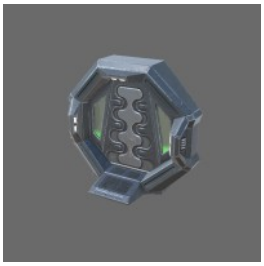
Prefabs	Tris (LOD 0)	Colors	Notes	Position
19 8	660 – 2270	5	*Customize the color of the main elements of the walls.*	Position X 0 y 0 z 0 Offset X 10 Y 10 Z 10

The corridors between the rooms and/or a web of tunnels.

DISPLAYS

Prefabs	Tris (LOD 0)	Colors	Notes	Position
26 19	2 – 18	1	*Each screen has its own independent material.. However there are the same in size screens, so You may exchange their materials.*	Free

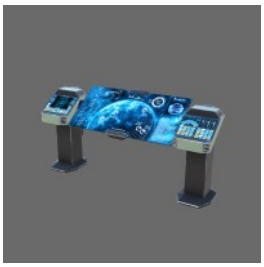
The Displays are possible to place on every appropriate surface, for example the walls. All Displays Prefabs are included in the Equipment Prefabs. The screens are animated.



DOORS

Prefabs	Tris (LOD 0)	Colors	Notes	Position
15 7	4 – 920	1		Position X 5 y 0 z 5 Offset X 10 Y 10 Z 10

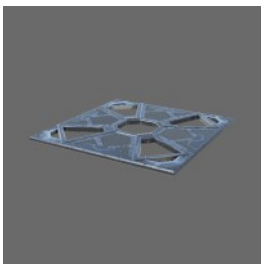
The Doors and Energy Gates for inside and outside. The special floor piece for the transport to move over. The railings are also available for the free positioning.



EQUIPMENT

Prefabs	Tris (LOD 0)	Colors	Notes	Position
29 14	76 – 6160	5	*Customize the color of the band of the stands.*	Free

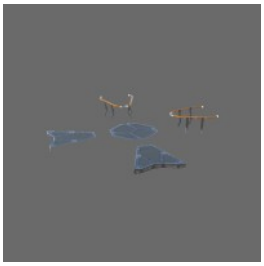
The Digital Equipment – from the little boxes, to the tables and the controlling door consoles – all with the animated displays. On how to setup consoles, please refer to “the Door and Consoles Setup”, in this documentation



FLOORS

Prefabs	Tris (LOD 0)	Colors	Notes	Position
31 19	128 – 592	1		Position X 0 y 0 z 0 Offset X 10 Y 10 Z 10

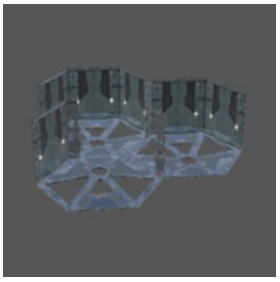
The different variations of the floors (and ceiling) pieces for small and large rooms. If the building is one-story-tall, pick the one-sided piece to save on triangles.



FLOORS FILL

Prefabs	Tris (LOD 0)	Colors	Notes	Position
37 22	6 – 17804	1		Position X 0 y 0 z 0 Offset X 10 Y 10 Z 10

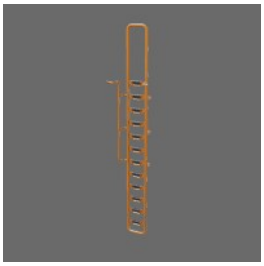
Plan the floors and ceilings in Your scene. Whether it be total fill of the surface, or some clear parts with railings, or the center piece removed for placing the ladder.



HEXA & PENTA ROOMS

Prefabs	Tris (LOD 0)	Colors	Notes	Position
5 0	3240 – 3840	1		Position X 0 y 0 z 0 Offset X 10 Y 10 Z 10

Vertical climbing on the walls outside or the ladder into the storage room. And who know where else these will simplify the way.



LADDERS

Prefabs	Tris (LOD 0)	Colors	Notes	Position
2 1	3240 – 3840	1		Position X 0 y 0 z 0 Offset X 10 Y 10 Z 10

Vertical climbing on the walls outside or the ladder into the storage room. And who know where else these will simplify the way.

PARTITIONS

Prefabs	Tris (LOD 0)	Colors	Notes	Position
7 7	1420 – 3192	5	*Customize the color of the warning stripes.*	Position X 0 y 0 z 0 Offset X 10 Y 10 Z 10 or Free

Made for the visual zoning of the room, Partitions may be placed using the recommended position or freely.

PARTITIONS 2

Prefabs	Tris (LOD 0)	Colors	Notes	Position
15 14	782 – 5526	5	*Customize the color of the vertical elements and the pipes itself.*	Position X 0 y 0 z 0 Offset X 10 Y 10 Z 10

Many Partitions 2 has horizontal and vertical pipelines. It enhances the industrial or bunker feeling, where appropriate.

PIPELINE

Prefabs	Tris (LOD 0)	Colors	Notes	Position
8 0	1336 – 1696	5	*Customize the color of the pipes.*	Free

Pipe alone, for making Your own pipelines. So it is possible combining them in length, making the lines of pipes for positioning for example horizontally along the walls.

PROPS

Prefabs	Tris (LOD 0)	Colors	Notes	Position
6 6	1948 – 4340	5	*Customize the color of the painted elements of the boxes and barrels.*	Free

Boxes, Barrels and Tanks for free positioning.

STAIRS

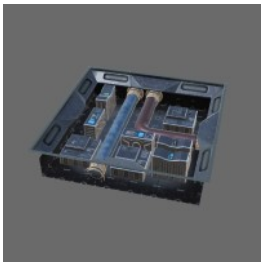
Prefabs	Tris (LOD 0)	Colors	Notes	Position
42 31	44 – 3146	1		Position X Free Y 0 Z Free Offset X 2 Y Free Z 2

The most hard asset for placing is the Stairs prefab. It require vertical adjustment by Y. But horizontal offset is 2m.
With independent pieces of the prefab Stairs, it is possible making not only the way up, but also various platforms and transitions with crossings.

SUPPORT

Prefabs	Tris (LOD 0)	Colors	Notes	Position
4 4	656 – 2624	1		Position X 0 y 0 z 0 Offset X 10 Y 10 Z 10

Made to look strong, they enhance the feel of heaviness and safety of the construction. It is possible not to use Support prefab.



TOP-BOTTOM

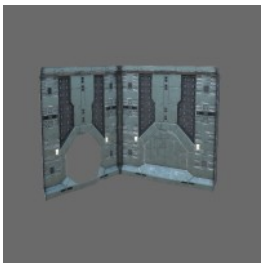
Prefabs	Tris (LOD 0)	Colors	Notes	Position
5 3	3550 – 5472	1		Position X 0 y 0 z 0 Offset X 10 Y 10 Z 10

The little details does matter. Placed at the floor/ceiling, Top-Bottom prefab is meant for enhancing the atmosphere, telling the different stories – like the area under maintenance or technical zone.

TOP-DOWN

Prefabs	Tris (LOD 0)	Colors	Notes	Position
22 8	68 – 1184	1		Position X 0 y 0 z 0 Offset X 10 Y 10 Z 10

Outside-styled walls and closing elements to make a scene for the Top-Down view.



WALLS

Prefabs	Tris (LOD 0)	Colors	Notes	Position
26 25	100 – 384	5	*Customize the color of the main elements of the walls.*	Position X 0 y 0 z 0 Offset X 10 Y 10 Z 10

The Wall Lights and Walls prefab. With/without the opening for placing the door. From one wall piece to four wall pieces combined.

And one **Zzz Point Light** scripted Prefab (AUGMENTED Version)

Heavy Station Kit HANGARS Prefabs

The Heavy Station Kit hangars 2.50 AUGMENTED has **282** Prefabs:
The Heavy Station Kit hangars 2.50 has **183** Prefabs:

AGGREGATES

Prefabs	Tris (LOD 0)	Colors	Notes	Align
25 11	70 – 18510	6	*Customize colors for some elements*	for Rails: Position X 5 y 0 z 5 Offset X 10 Y 10 Z 10 for Other: FREE

Aggregates
Ballons and Cables, Cargocase, Consoles, Rail and Crane modular system, Reactor and Server.

AGGREGATES 2

Prefabs	Tris (LOD 0)	Colors	Notes	Align
13 0	958 – 12056	1		FREE

Aggregates2
are huge single, dual and trio Pipes, Flat reactor, Huge barrels with various pipes.

ARCHES

Prefabs	Tris (LOD 0)	Colors	Notes	Align
26 18	420 – 716	2	*Customize colors for inside panels*	Position X 5 y 0 z 5 Offset X 10 Y 10 Z 10

Arches
are three types of L shaped design elements, with customizable solid and/or transparent pieces .

DISPLAYS

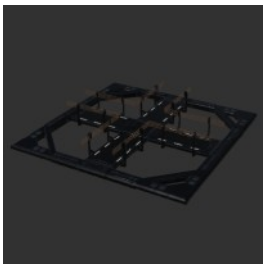
Prefabs	Tris	Colors	Notes	Align
7 6	2 – 24	1		Parent Object

DOORS

Prefabs	Tris	Colors	Notes	Align
5 5	78 – 1236	6	*Customize colors for some elements*	Position X 5 y 0 z 5 Offset X 10 Y 10 Z 10

ELEVATORS

Prefabs	Tris	Colors	Notes	Align
1 1	9944	1		Position X 0 y 0 z 0 Offset X 10 Y 10 Z 10



FLOORS

Prefabs	Tris (LOD 0)	Colors	Notes	Align
66 64	28 – 5120	2	*Customize colors for fills elements*	Position X 0 y 0 z 0 Offset X 10 Y 10 Z 10

Floors
come as Floor Frames in sizes of 10 and 5 meters. There are also two types of narrow Transition elements and four types of Hand-rails. To increase visual interest, there are solid and transparent Floor Fill pieces to fit in floor frames.

GARAGE

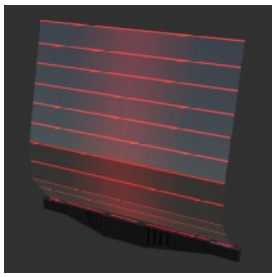
Prefabs	Tris (LOD 0)	Colors	Notes	Align
8 0	60 – 2374	1		FREE

Garage
are modular pieces for vehicle with wheels. Around that can be placed special maintenance platforms and with ladder for humans to get up. Also theme has own console and aggregate.

GATEWAY

Prefabs	Tris (LOD 0)	Colors	Notes	Align
20 15	44 – 10174	1		FREE

Gateway
are room-scaled areas for vehicles, with full-sized animated gates.



OUTSIDE

Prefabs	Tris (LOD 0)	Colors	Notes	Align
10 4	178 – 15746	1		FREE

Outside
has modular energy barrier with intent for placing around the base. Also huge stairs, animated radar, cone-shaped station.

PIPELINE 1

Prefabs	Tris (LOD 0)	Colors	Notes	Align
5 0	364 – 728	6		FREE

Pipeline1
are new small modular pipelines.

PIPELINE 2

Prefabs	Tris (LOD 0)	Colors	Notes	Align
19 0	928 – 2292	6		FREE

Pipeline2
looks like ones in Base v2, but now they are modular.

PROPS

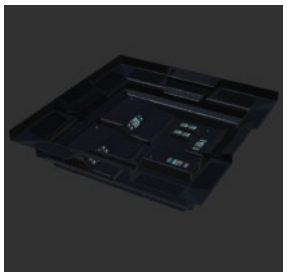
Prefabs	Tris (LOD 0)	Colors	Notes	Align
8 6	236 – 3852	6	*Customize colors for some elements*	FREE

Props
are small and medium barrels, small battery, small to medium to big boxes, and small but narrow and long box that can be stacked on itself in pyramid form.

SUPPORTS

Prefabs	Tris (LOD 0)	Colors	Notes	Align
19 15	132 – 4336	1		Position X 0 y 0 z 0 or FREE Offset X 10 Y 10 Z 10 and FREE

Supports
are used to enhance heavy look of the base, and they fit into special slot in Floor Frame pieces.



TOP BOTTOM

Prefabs	Tris (LOD 0)	Colors	Notes	Align
7 7	564 – 574	2	*Customize colors for some elements*	Position X 0 y 0 z 0 Offset X 10 Y 10 Z 10

Top Bottom
are used to increase visual depth of the level when needed, and they come in three different pieces.

TOP-DOWN

Prefabs	Tris (LOD 0)	Colors	Notes	Align
15 15	74 – 370	1		Position X 0, 5 y 0, 5 z 0, 5 Offset X 10, 5 Y 10, 5 Z 10, 5 and FREE

Top-Down
has four types of walls for outside, and elements to close gaps for Top-Down use.

TOP-DOWN 2

Prefabs	Tris (LOD 0)	Colors	Notes	Align
9 0	164 – 1852	1		Position X 0, 5 y 0, 5 z 0, 5 Offset X 10, 5 Y 10, 5 Z 10, 5 and FREE

Top-Down2
has new supports for outside that strenghten visual look, three additional walls for outside, and pieces for Top-Down use.

WALLS

Prefabs	Tris (LOD 0)	Colors	Notes	Align
18 16	111 – 534	6	*Customize colors for some elements*	Position X 0 y 0 z 0 Offset X 10 Y 10 Z 10 and FREE

Walls has 5 and 10 metres elements, flat and L and C shaped, with openings for doors, gates and windows.

And one **Zzz Point Light** scripted Prefab (AUGMENTED Version)

Heavy Station Kit COLONY Prefabs

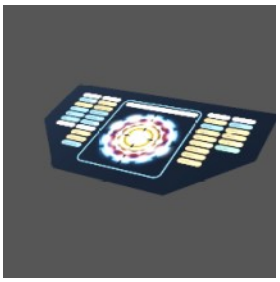
The Heavy Station Kit colony 2.50 AUGMENTED has **451** Prefabs:
The Heavy Station Kit colony 2.50 has **291** Prefabs:

DECORATIONS

Prefabs	Tris (LOD 0)	Colors	Notes	Position
36 13	120 – 15000	1		Free

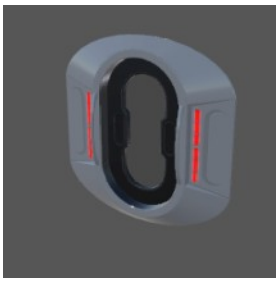
DEVICES

Prefabs	Tris (LOD 0)	Colors	Notes	Position
15 9	46 – 956	6	*Customize colors for some elements*	Free



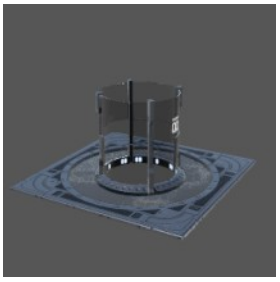
DISPLAYS

Prefabs	Tris (LOD 0)	Colors	Notes	Position
19 12	2 – 8	1		Free



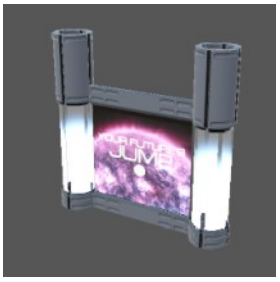
DOOR_WINDOW

Prefabs	Tris (LOD 0)	Colors	Notes	Position
13 8	302 – 2496	6	*Customize colors for Emission Color*	Position X 5 y 0 z 5 Offset X 10 Y 10 Z 10



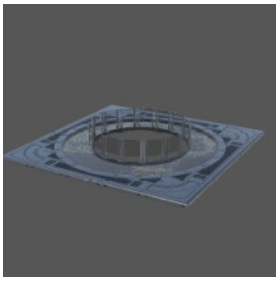
ELEVATOR

Prefabs	Tris (LOD 0)	Colors	Notes	Position
14 9	60 – 5760	1		Position X 0 y 0 z 0 Offset X 10 Y 10 Z 10



EQUIPMENT

Prefabs	Tris (LOD 0)	Colors	Notes	Position
31 20	134 – 5800	1		Free

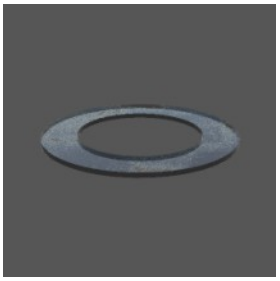


FLOORS

Prefabs	Tris (LOD 0)	Colors	Notes	Position
41 34	14 – 1042	1		Position X 0 y 0 z 0 Offset X 10 Y 10 Z 10

FURNITURE

Prefabs	Tris (LOD 0)	Colors	Notes	Position
43 26	76 – 4288	6	*Customize colors for some elements*	Free



GLASS

Prefabs	Tris (LOD 0)	Colors	Notes	Position
17 17	4 – 5600	1		Free

KITCHEN

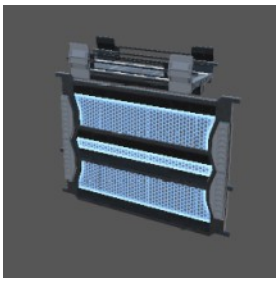
Prefabs	Tris (LOD 0)	Colors	Notes	Position
60 34	10 – 2680	6	*Customize colors for some elements*	Free

OBJECTS

Prefabs	Tris (LOD 0)	Colors	Notes	Position
22 11	80 – 3140	1	*Customize the color of the vertical elements.*	Free

OUTSIDE_TOPDN

Prefabs	Tris (LOD 0)	Colors	Notes	Position
33 19	30 – 3076	1	*Customize the color of the vertical elements.*	Position X 0 y 0 z 0 Offset X 10 Y 10 Z 10



OUTSIDE_TOPDN_2

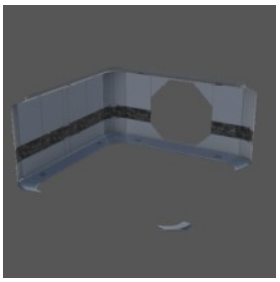
Prefabs	Tris (LOD 0)	Colors	Notes	Position
24 14	56 – 3652	1		Position X 0 y 0 z 0 Offset X 10 Y 10 Z 10

STAIRS

Prefabs	Tris (LOD 0)	Colors	Notes	Position
16 7	76 – 11926	1		Position X 0 y 0 z 0 Offset X 10 Y 10 Z 10

VENTILATION

Prefabs	Tris (LOD 0)	Colors	Notes	Position
13 12	2 – 7690	6	*Customize colors for Emission*	Position X 0 y 0 z 0 Offset X 1 Y 1 Z 1



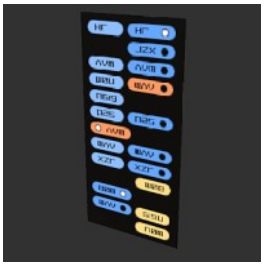
WALLS

Prefabs	Tris (LOD 0)	Colors	Notes	Position
53 46	14 – 576	6		Position X 0 y 0 z 0 Offset X 10 Y 10 Z 10

And one **Zzz Point Light** scripted Prefab (AUGMENTED VErsion)

Heavy Station Kit DUGOUT Prefabs

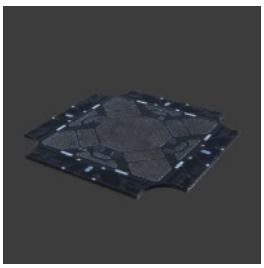
The Heavy Station Kit dugout AUGMENTED has **621** Prefabs:
The Heavy Station Kit dugout has **489** Prefabs:



DISPLAYS

Prefabs	Tris	Materials
42 26	2 – 852	Heavy Station Kit / DUGOUT / Materials / N_Screens_A Heavy Station Kit / DUGOUT / Materials / N_Screens_BC Heavy Station Kit / DUGOUT / Materials / N_Screens_S **(Augmented only)** Heavy Station Kit / DUGOUT / Materials / N_Nozzle **(Augmented only)**

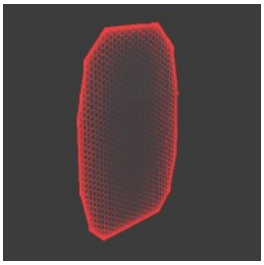
Animated displays are part of prefabs. Flats can be used apart.



GAPS

Prefabs	Tris	Materials
22 22	2 – 280	Heavy Station Kit / DUGOUT / Materials / N_Gaps Heavy Station Kit / DUGOUT / Materials / N_A_Out Heavy Station Kit / DUGOUT / Materials / N_A_In

See-through Floor pieces.



PLATES

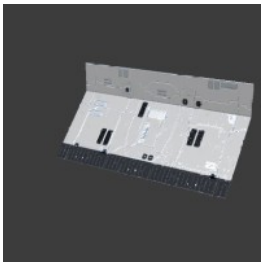
Prefabs	Tris	Materials
28 0	2 – 14	Heavy Station Kit / DUGOUT / Materials / N_Screens_P **(AUGMENTED ONLY)**

Shop signs, Transparent Barriers, etc.

TYPE A Exterior

Prefabs	Tris	Materials
57 57	10 – 2852	Heavy Station Kit / DUGOUT / Materials / N_A_Out

Biggest parts. Windows, External props, etc. Core, Transition and Props.



TYPE A Interior

Prefabs	Tris	Materials
53 53	3 – 1712	Heavy Station Kit / DUGOUT / Materials / N_A_In

Biggest parts. Windows, big Crossroads, wall hatches, Stairs, etc. Core and Transition.

TYPE A Props

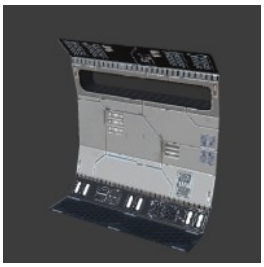
Prefabs	Tris	Materials
35 35	50 – 4328	Heavy Station Kit / DUGOUT / Materials / N_A_Props

Interior arches, Engine, Stands, Pipelines, etc. Interior & Exterior Props.

TYPE B Exterior

Prefabs	Tris	Materials
37 37	24 – 112	Heavy Station Kit / DUGOUT / Materials / N_BC_Out

B Exterior Unique Windows, Floor Pipeline Socket. Core, Transition, etc.



TYPE B Interior

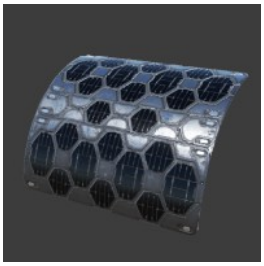
Prefabs	Tris	Materials
45 45	2 – 508	Heavy Station Kit / DUGOUT / Materials / N_BC_In

B Interior Unique Windows, Floor Pipeline Socket. Core and Transition.

TYPE B Props

Prefabs	Tris	Materials
17 17	262 – 1984	Heavy Station Kit / DUGOUT / Materials / N_BC_Props

Wall point devices, pipelines, handrails. Interior Props.



TYPE BC Exterior

Prefabs	Tris	Materials
99 99	1 – 3152	Prefabs/Walls/Meshes/Materials/N_BC_Out

Applies also for B and C. Core, Transition and Props.

TYPE BC Interior

Prefabs	Tris	Materials
15 15	14 – 1402	Heavy Station Kit / DUGOUT / Materials / N_BC_In

B&C joint Pieces, tiny Crossroads, floor hatches. Transition.

TYPE BC Props

Prefabs	Tris	Materials
4 4	310 – 2110	Heavy Station Kit / DUGOUT / Materials / N_BC_Props

Switches, Ladder and Fan. Interior Props.

TYPE C Exterior

Prefabs	Tris	Materials
24 24	32 – 72	Heavy Station Kit / DUGOUT / Materials / N_BC_Out

C Exterior Unique Windows. Core, Transition.

TYPE C Interior

Prefabs	Tris	Materials
36 36	2 – 297	Heavy Station Kit / DUGOUT / Materials / N_BC_In

Core and Transition.

TYPE C Props

Prefabs	Tris	Materials
11 11	288 – 1652	Heavy Station Kit / DUGOUT / Materials / N_BC_Props

Wall point devices, pipelines, handrails. Interior Props.

TYPE S

Prefabs	Tris	Materials
35 6	80 – 1672	Heavy Station Kit / DUGOUT / Materials / N_S_Out_In

Cockpit, external equipment, etc. Exterior & Interior Core, Transition and Props.

TYPE S Props

Prefabs	Tris	Materials
61 2	142 – 5592	Heavy Station Kit / DUGOUT / Materials / N_S_Props **(AUGMENTED ONLY)**

Living, consoles, etc. Interior props.

Glass folder in Prefabs is for source meshes only. Complete items are in their appropriate Themes.

Heavy Station Kit DUGOUT Blueprints

Blueprints are pieces of exterior, interior and props put together to make a rough blockout faster.
Comes with AUGMENTED version of Heavy Station Kit dugout.

The Heavy Station Kit dugout 1.03 AUGMENTED has **131** Blueprints:
The Heavy Station Kit dugout 1.03 has **110** Blueprints:

Shuttle Systems

Blueprints	Purpose
15 0	Movable and controllable pieces like Engines that come with Scripts

Type_A

Blueprints	Purpose
27 27	Exterior, interior and props Type A Blueprint pieces.

Type_B

Blueprints	Purpose
52 52	Exterior, interior and props Type B Blueprint pieces.

Type_C

Blueprints	Purpose
31 31	Exterior, interior and props Type C Blueprint pieces.

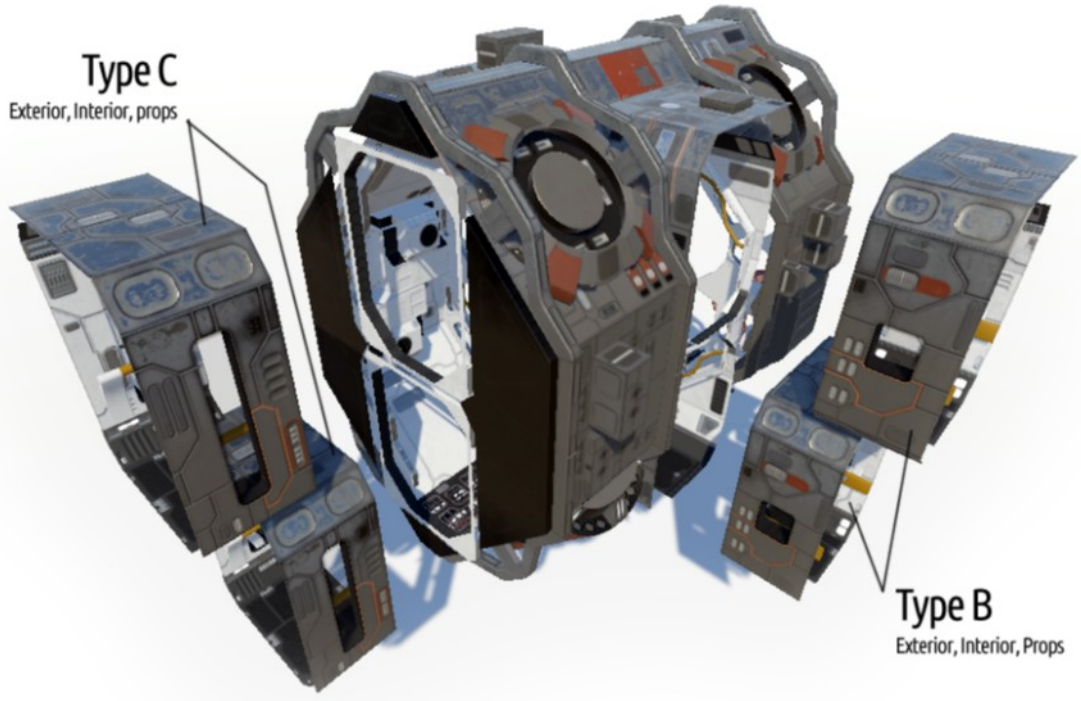
Type_S

Blueprints	Purpose
6 0	Exterior, interior and props Type S Blueprint pieces.

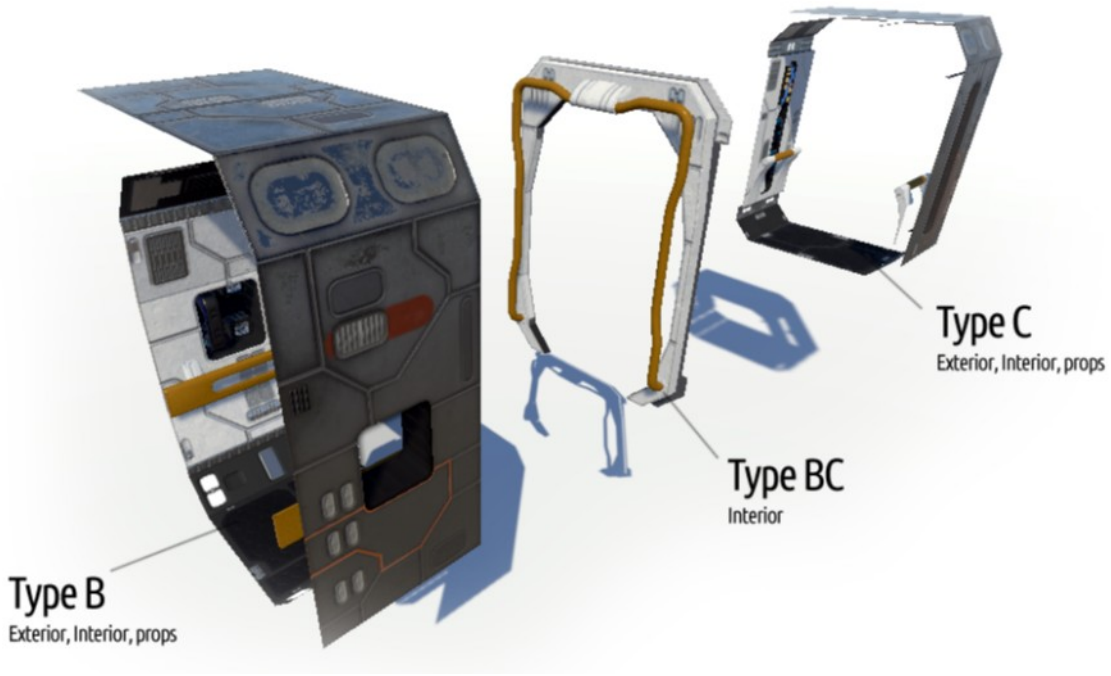
Blueprint setup

Type A Compilation

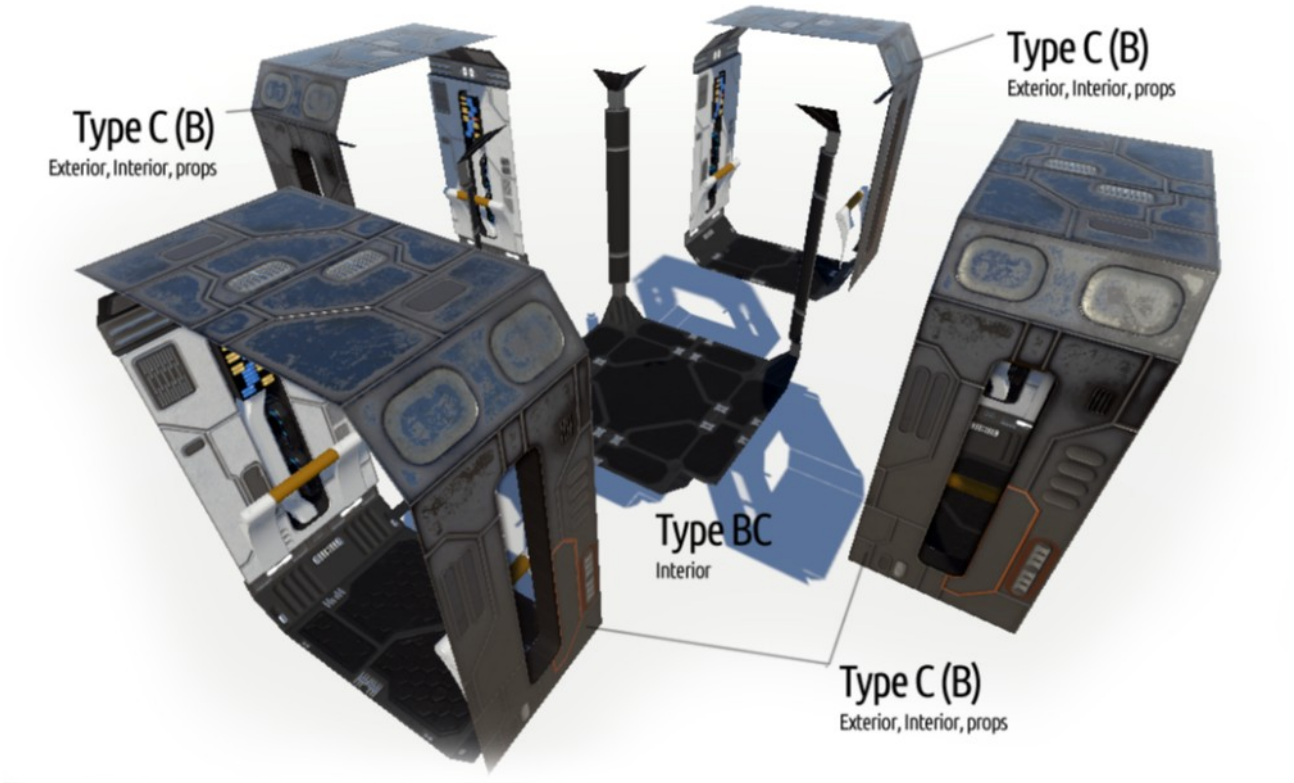
Type A, B, C Compilation



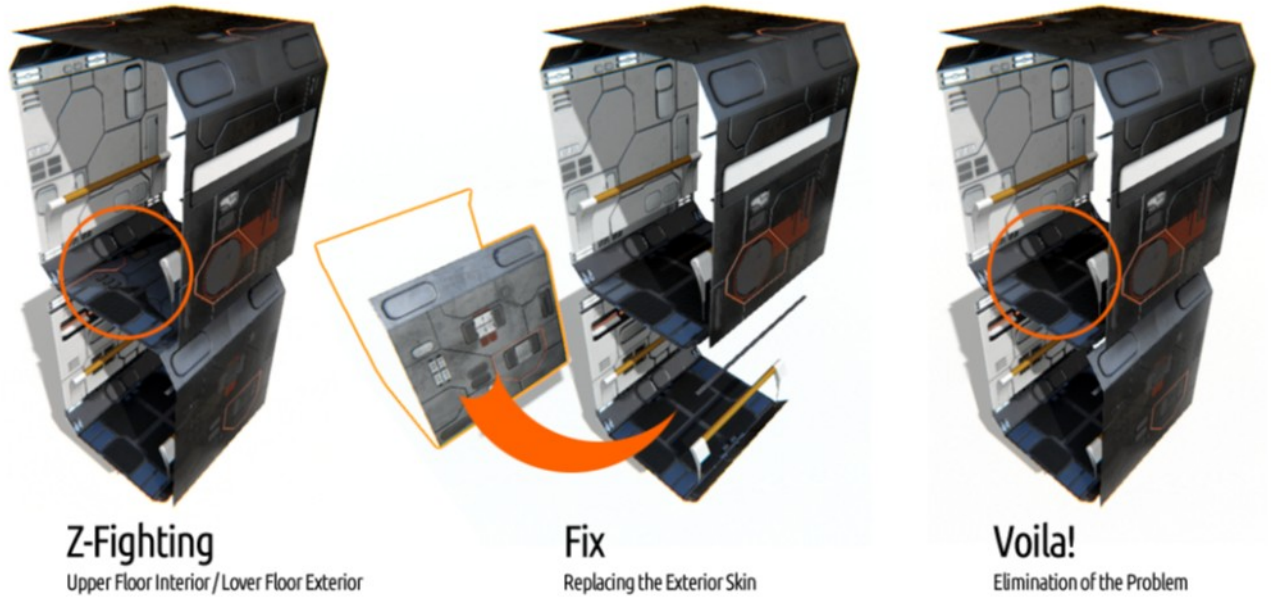
Type B&C Joint



Types B&C Crossroad



Types B&C Z-Fighting Fixed



When you are making multi floor section using B or C pieces ‘Z-Fighting’ will occur, as exterior of lower floor section will creep in.

It is easy to solve. Hide this part from Blueprint and add an appropriate piece without exterior to replace it (from Prefab folder).

‘Z-Fighting’ vanished as error introducing piece got replaced.

Heavy Station Kit BASE Materials

DISPLAYS (Materials)	(Meshes)
B2_Eq1	B2_Eq_1
B2_Eq2	B2_Eq_2
B2_Eq3	B2_Eq_3
B2_Eq5A1	B2_Eq_5D, B2_Eq_5T
B2_Eq5A2	B2_Eq_5D. B2_Eq_5T
B2_Eq41	B2_Eq_4, B2_Eq_7
B2_Eq42	B2_Eq_4, B2_Eq_7
B2_Eq43	B2_Eq_4, B2_Eq_7
B2_Eq44	B2_Eq_4, B2_Eq_7
B2_Eq51	B2_Eq_5, B2_Eq_7
B2_Eq52	B2_Eq_5, B2_Eq_7
B2_Eq_23c	B2_EQ_23c
B2_TB_Med	B2_TB_M1, B2_TB_M2, B2_TB_M3
B2_TB_Small	B2_TB_S1, B2_TB_S2, B2_TB_S3
B2_TD_Part2	B2_TD_PRT2
Disp_Cons	B2_Disp_Cons
Disp_Cons_Mode	B2_Disp_Cons
Disp_Cons_Power	B2_Disp_Cons

TOP-BOTTOM (Materials)	(Meshes)
B2_Top_Bottom	TB_(1-3)_F
B2_TB_PH	TB_Hexa_F, TB_Penta_F

TOP-DOWN (Materials)	(Meshes)
B2_TD_2_RGlass	TD_base2_RGlass_(1-3)
B2_TD_Roof	TD_base_Roof(1-3)
B2_TD_Roof1	TD_base2_RoofG_(1-3)
B2_TopDown	TD_Base2_WE_M, TD_base_part2, TD_base_outwall(...), TD_base_topwall, TD_base_topwall1, TD_base_topwall2, Base2_Egate_0, TD_base_topwall_(2...-4), Ladder2, B2_TD_HandRail, B2_TD_HandRail2
B2_TopDown1	Arches_C_1, TD_base_topwall3, Floor_5_base, Floor_5_base_C, Floor_5_base_F, Floor_5_base_Plate, Floor_5_base_TB, B2_Egate, B2_EGate0, B2_EgateA, B2_EgateB, Base2_EGate_1

DOORS (Materials)	(Meshes)
B2_EG_OFF	B2_EG
B2_EG_ON	B2_EG
Glass_Dark	Door_a_glass
Glass_Green	Door_a_glass
Glass_Red	Door_a_glass

EQUIPMENT (Materials)	(Meshes)
B2_Eq(0-4)	Eq(1-5), Eq(8-10)
	Chan_(11-12), Chan_(41-44), Arm
	Door_a, Door_a_H, Door_a_slide
B2_Eq_Out	Eq20a, Eq20b, Eq20c, Eq21, Eq23c
B2_Eq_Out1	Eq23, Eq23a, Eq23b, Eq23d

FLOORS (Materials)	(Meshes)
B2_Floors	Floor_(1...-4…), Floor_(6...-7…)
	FBC_(…), FCC_(…), FLC_(...)
	FF_base2_El(...)
	FF_base2_Rel(...)
	St_Railing(...)
	B2_fba
	Support_(1-4)
B2_Floors_PH	Floor_Hexa(…), Floor_Penta(...)
	B2_FFH(…), B2_FFP(...)

PROPS (Materials)	(Meshes)
B2_Props(0-4)	Barel(1-2), Box(1-2), Tank(1-2)

STAIRS (Materials)	(Meshes)
B2_Stairs, B2_Stairs_NL	St_1(…)-St_10(...)
	B2_HR_(...)
	ladder1

WALLS (Materials)	(Meshes)
B2_Walls(0-4)	All Arches, Partitions, Partitions2, Pipeline, and Walls Meshes

Heavy Station Kit HANGARS Materials

AGGREGATES (Materials)	(Meshes)
H2_Aggregates_(0-5)	Agg_(…), H2_P1_(1-5), H2_P2_(01-19)

AGGREGATES2 (Materials)	(Meshes)
H2_Agg2_Light	H2_Agg_Light1, H2_Agg_Light2
H2_Aggregates	H2_Agg_(1-13)

DISPLAYS (Materials)	(Meshes)
H2_Dis_Door	H2_Dis_Doors
H2_Disp_Cons1	H2_Cons1
H2_Disp_Cons1D	H2_Cons1_D
H2_Disp_Cons2	H2_Cons2
H2_Disp_Cons4	H2_Cons4
H2_Disp_Cons4D	H2_Cons4_D
H2_Disp_Garage	H2_Gar_5_Light

DOORS (Materials)	(Meshes)
Door2_(Green, Grey, Red)	door_3_glass

PROPS (Materials)	(Meshes)
H2_Props_(0-5)	hangar_barrel(1-2), hangar_battery1, hangar_box(1-5)

SUPPORTS (Materials)	(Meshes)
H2_Supp_Doors_(0-5)	H2_Cel_(02, 04), H2_Cel_(1-8), H2_Sup_(1-4), Mount, SG_2, Support_5(…), Door_(0-3), Gate, Elevator1(…), H2_Elevator, Plate

WALLS (Materials)	(Meshes)
H2_Arches(...)	Glass_Arch_(1-4), Glass_Out_(…), Glass_Wall_(…)
H2_Walls1_(0-5)	Wall_(1-4), arche_(1-3)
H2_Walls2_(0-5)	Wall_(5-7)

FLOORS (Materials)	(Meshes)
H2_Floors, H2_Floors_NoL	floor_0h(…), floor_1h(…), floor_1h_fill(…), floor_1h_HR, floor_2h(…), floor_2h_fill(…), floor_2h_HR, floor_3h(…), floor_4h(…), floor_5h(…), H2_floor_6(…), handrail_(1-4), transition(1-3)
H2_Floors_NoL, H2_FloorsGlass	floor_1h_Glass, floor_1h_Glass_one, floor_2h_Glass, floor_2h_Glass_one, H2_FF_T

GARAGE (Materials)	(Meshes)
H2_Garage	H2_Gar_(1-8)

GATEWAY (Materials)	(Meshes)
H2_Gateway	Brace, Gateway(1-2), Hook, GW_gate(1-2), Ladder
H2_Shield	Shield

OUTSIDE (Materials)	(Meshes)
H2_LightWall, H2_LightWall_2	H2_Out_Ewall(...)
H2_Out1_Light	H2_Out_1_Light
H2_Out2_Light	H2_Out_2A_Light
H2_Out3_Light	H2_Out_3_Light, H2_Out_3A_Light
H2_Outside	H2_Out_(1-6)

TOP BOTTOM (Materials)	(Meshes)
H2_TB, H2_TB_A	H2_TB_Cover, TB_II_(1-3)

TOP-DOWN (Materials)	(Meshes)
H2_Top-Down	TD_hangar_ARCH(0-1), TD_hangar_OW(1-4), TD_hangar_TW(0-2)

TOP-DOWN2 (Materials)	(Meshes)
H2_TopDown2	H2_Outwall(…), H2_Support(…), H2_Topwall

Heavy Station Kit COLONY Materials

DECORATION (Materials)	(Meshes)
C2_Decoration	C2_Dec_PAN(1-3),
	C_Nat_Fern(…), C_Nat_Flower(…),
	C_Nat_Grass(…), C_Nat_Ground(…)
C2_Stones	C2_Dec_PAN1_St, C2_Dec_PAN2_St,
	C_Stone(2-4)
C_Leaf(1-3)	C_Tree_Leafs1
C_Tree(1-3)	C_Tree_Tree1

DEVICES (Materials)	(Meshes)
C2_Devices_(0-5)	C_Dev_Bidet, C_Dev_Bowl,
	C_Dev_Button(1-3), C_Dev_Console,
	C_Dev_Pallet, C_Dev_Pod_Med,
	C_Dev_Podium(1-3), C_Dev_Sho,
	C_Dev_Sup_Med, C_Dev_Support,
	C_Dev_Tap, C_Dev_Taps,
	C_Dev_Towel, C_Dev_Uri,
	C_Dev_Washstand, C_Pot(1-3),
	C_Dev_Washstand1

FLOORS (Materials)	(Meshes)
C2_Floors(...)	C_Floor(…), C2_Floor(...)
	C2_Floor_HR1, C2_Floor_HR1A,
	C_Stairs3_A1m, C_Stairs3_Am,
	C_Stairs3_B1m, C_Stairs3_Bm,
	C_Stairs3_C1m, C_Stairs3_Cm,
	C_Stairs_1, C_Stairs_2, C_Stairs_2A,
	C_Stairs_2B, C_Stairs_2C,
	C_Stairs_2D, C_Stairs_3, C_Stairs_4

KITCHEN (Materials)	(Meshes)
C2_Objects2_(0-5), NH	C2_Blender, C2_Bracket(…),
	C2_CoffeeTable, C2_Container(1-4),
	C2_Dish(1-5), C2_Drawer(30,60),
	C2_Fork(1-3), C2_Jalousie(…),
	C2_Knife(1-4), C2_Label_(…),
	C2_Lamp, C2_Microwave,
	C2_MicrowaveDoor, C2_Partition_D,
	C2_Partition_U, C2_Scales(…),
	C2_Screen, C2_Shelf(…), C2_Sound,
	C2_Spiracle(…), C2_Table(…),
	C2_Teapot(…), C2_Toster,
	C2_WashingMachine, C2_Work(...)

OBJECTS (Materials)	(Meshes)
C2_Objects	C_Obj_Bottle, C_Obj_Bottles,
	C_Obj_Camera, C_Obj_Container,
	C_Obj_Containers, C_Obj_Cutlery2,
	C_Obj_Fork, C_Obj_Hold_A,
	C_Obj_Holder(1-2), C_Obj_Knife,
	C_Obj_Microscope, C_Obj_Monitor,
	C_Obj_Microscope1, C_Obj_Netbook,
	C_Obj_Pad, C_Obj_Pen, C_Obj_Tools,
	C_Obj_Spoon, C_Obj_Thermos,

DISPLAYS (Materials)	(Meshes)
C2_ServerL, C_Light, C_Energy_Door	C_Dis_Stand2
C2_ServerS, C_Light, C_Energy_Door	C_Dis_Stand1
C_Control1	C_Dis_Control1
C_Control2	C_Dis_Control2
C_Control3	C_Dis_Control3
C_Displays(1-4)	C_Dis_Planet, C_Dis_Scr(2-3)
C_ElevDispDn, C_ElevDispMove	C_El_DisplDn
C_ElevDispUp	C_El_DisplUp
C_Med	C_Dis_MConsole
C_Monitor	C_Dis_Monitor
C_Netbook	C_Dis_Netbook
C_Pad	C_Dis_Pad
C_Ray	C_Dis_Scr1(…), C_Dis_Ray

DOOR-WINDOW (Materials)	(Meshes)
C2_Doors(...)	C_BorderKit, C_Door(…), C_Vent(…),
	C_Win(...)

ELEVATOR (Materials)	Meshes
C2_Elevator	C2_El_Cons2, C2_El_Elevator(...),
	C2_El_Tube, C2_El_Wall,
	C_El_Cabine, C_El_Console,
	C_El_Flloor(…), C_El_Plate(…),
	C_El_Support, C_El_Tank

FURNITURE (Materials)	(Meshes)
C2_Furniture_(0-5)	C_Fu_(…)

GLASS (Materials)	(Meshes)
C_Mirr	C_Dis_Mirror
C_Glass	C_Glass_HR(…), C_Glass_Sho,
	C_GlassD&W, C_GlassPano,
	C_GlassSlider, Glass_Door,
	Glass_El_Floor, Glass_El_FloorH(…),
	Glass_P1, Glass_R1, Glass_R2,
	Glass_Stairs(1-3), Glass_StairsB,
	Glass_StairsC, Glass_StairsD,
	Glass_Table(1-3), Glass_Trans(2-3),
	Glass_Wall_2m, Glass_Wall_3m,
	Glass_Window
	C_Stairs3_A1g, C_Stairs3_Ag,
	C_Stairs3_B1g, C_Stairs3_Bg,
	C_Stairs3_C1g, C_Stairs3_Cg
	C2_Blender_Glass, C2_Dish3_Glass,
	C2_Table_Door_Glass, C2_WM_Glass,
	C2_MicrowaveGlass, C_Obj_Cup,
	C_Obj_TestTube(...)
C_El	Glass_Elev, Glass_Elev1
C_Glass_0, C_Glass_min(...),	Glass_Elevator
C_Glass_plus(...)	

OUTSIDE-TOPDN-2 (Materials)	(Meshes)
C2_Outside2, C2_Outside2_Z	C2_Out2(...)

	C_Obj_Thermoses, C_Obj_Tray

OUTSIDE-TOPDN (Materials)	(Meshes)
C2_Misc	C_Misk_Aerial, C_Misk_AerialBase, C_Misk_FoSup, C_Misk_SolarHolder, C_Misk_Foundation(…), C_Misk_GlassWallCorner, C_Misk_SolarPanel
C2_Outside_(1-2)	C_Out_Support, C_Out_TD_(1-4), C_Out_Trans_(…), C_Out_Wall_(...)

WALLS (Materials)	(Meshes)
C2_Walls_(0-5)	C_1Walls(…), C_2Walls(…), C_3Walls(…), C_4Walls(…), C_Arche_(1-2), C_Walls_1, C_Wall_Part_(…)

Heavy Station Kit DUGOUT Materials

DUGOUT / Materials / ...	DUGOUT / Prefabs / Displays / Meshes / ...
N_Screens_A	N_Screens_A...
N_Screens_BC	N_Screens_B..., N_Screens_C...
N_Nozzle	N_Screens_S_15
N_Screens_S	N_Screens_S...

	DUGOUT / Prefabs / Gaps / Meshes / ...
N_Gaps	N_Gaps_A..., N_Gaps_B..., N_Gaps_Unit
N_A_Out	N_Solid_A, N_Solid_A_C, N_Solid_A_F
N_A_In	N_Solid_A...

	DUGOUT / Prefabs / Glass / Meshes / ...
N_Glass	N_G...

	DUGOUT / Prefabs / Plates / Meshes / ...
N_Screens_P	N_DP...

	DUGOUT / Prefabs / Type A Exterior / Meshes / ...
N_A_Out	N_Ae...

	DUGOUT / Prefabs / Type A Interior / Meshes / ...
N_A_In	N_Ai..., NA_Pipeline...

	DUGOUT / Prefabs / Type A Props / Meshes / ...
N_A_Props	N_Ao..., N_D..., N_Crystal...

DUGOUT / Materials / ...	DUGOUT / Prefabs / Type B Exterior / Meshes / ...
N_BC_Out	N_B...

	DUGOUT / Prefabs / Type BC Exterior / Meshes / ...
N_BC_Out	N_BC...

	DUGOUT / Prefabs / Type C Exterior / Meshes / ...
N_BC_Out	N_C...

	DUGOUT / Prefabs / Type B Interior / Meshes / ...
N_BC_In	N_Bi...

	DUGOUT / Prefabs / Type BC Interior / Meshes / ...
N_BC_In	N_BCi...

	DUGOUT / Prefabs / Type C Interior / Meshes / ...
N_BC_In	N_Ci..., N_C_End_Pipeline

	DUGOUT / Prefabs / Type B Props / Meshes / ...
N_BC_Props	N_Bo...

	DUGOUT / Prefabs / Type BC Props / Meshes / ...
N_BC_Props	N_BCo...

	DUGOUT / Prefabs / Type C Props / Meshes / ...
N_BC_Props	N_Co

	DUGOUT / Prefabs / Type S / Meshes / ...
N_S_Out_In	N_Sg...

	DUGOUT / Prefabs / Type S Props / Meshes / ...
N_S_Props	N_So...

SCRIPTS

Customize Prefabs (scripts settings)

General info

All asset classes placed in common namespace **DotTeam.HSK**.
All script files are located in the corresponding subfolders of the **Assets > Heavy Station Kit > _common > Scripts** folder.

Doors & Gate2

Refers to prefabs	
HSK Base	**B2_Door** *Assets > Heavy Station Kit > BASE > Prefabs > Doors*
HSK Colony	**C2_Door** *Assets > Heavy Station Kit > COLONY > Prefabs > Door_Window*
HSK Hangars	**H2_Door, H2_Gate2** *Assets > Heavy Station Kit > HANGARS > Prefabs > Doors*
HSK Dugout	**DBCI_II_Door_HSK** *Assets > Heavy Station Kit > DUGOUT > Prefabs > Type BC Interior*

Door / **Gate2** Prefabs allows switching the operating modes of the door/gate in Edit and Game modes via public property ***Mode*** of **DotHskDoor** Script component attached to top-most Prefab game object, including:

Active	the door/gate is opening and closing automatically, at the approaching of a Player (gates are manually operated using the console). Initially, the door/gate is closed. Sound is being played, and opening and closing sounds of the panel sliding differ
Active Open	before the first pass, the doors/gates remain open (gate initially is open), after which the doors/gates continue to work in the same way as in Active mode
Blocked	the door/gate is closed. Sound of "the closed door" is being played, at approaching of a Player
Inactive Open	the door/gate is disabled, being fully open
Inactive Closed	the door/gate is disabled, being fully closed
Broken Open	the door/gate is disabled, being almost fully open
Broken Closed	the door/gate is disabled, being almost fully closed

Selecting of the door/gate operating mode is instant (happening immediately). In the Game mode the doors are automatically triggered when the character approaches.

*Useful public properties of **DotHskDoor** class*	
dotHskDoorMode **mode**	Allows set/read door operating mode, setting mode is instant - happening the next Update cycle. Acceptable values are ***dotHskDoorMode.{mode_id},*** where ***mode_id*** is one of following literals: ***active***, ***blocked***, ***inactiveOpen***, ***inactiveClosed***, ***brokenOpen***, ***brokenClosed*** (see description of operating modes above).
*Gate2 prefab only: **DotHskDoorHangarsGate2Console** script (attached to Console_Trigger GameObjects, childs of Console1 and Console2 GameObjects)*	
Texture **Banner**	On-screen hint image (source file **HSK_Gui.psd** included in *Assets > Heavy Station Kit > _common > Textures > GUI*)

Gate

Refers to prefabs	
HSK Hangars	GW_LargeGate, GW_SmallGate *Assets > Heavy Station Kit > HANGARS > Prefabs > Gateway*

Gate Prefab allows switching the operating modes of the gate in Edit and Game modes via public property ***Mode*** of **DotHskGate** Script component attached to the top-most Prefab game object. Gate prefab operates in the same manner as Gate2 prefab (see "Doors & Gate2" section) and its operational modes are including the same values as Gate2.

*Useful public properties of **DotHskGate** class*	
dotHskGateMode **mode**	Allows set/read gate operating mode, setting mode is instant - happening the next Update cycle. Acceptable values are ***dotHskGateMode.{mode_id},*** where ***mode_id*** is one of the following literals: ***active***, ***blocked***, ***inactiveOpen***, ***inactiveClosed***, ***brokenOpen***, ***brokenClosed*** (see list of operating modes in "Doors & Gate2" section).
bool ***isFullyOpen***	Equals ***true*** if the Gate is completely open at this time, otherwise - ***false***
bool ***isFullyClosed***	Equals ***true*** if the Gate is completely closed at this time, otherwise - ***false***
bool ***isStopped***	Equals ***true*** if the Gate is not moving at this time, otherwise - ***false***
DotHskGateHangarsConsole *script (attached to Console_Trigger GameObjects, childs of Console1 and Console2)*	
Texture **OpenTip**, **CloseTip**	On-screen hint images (source file **HSK_Gui.psd** included in *Assets > Heavy Station Kit > _common > Textures > GUI*)

Door's Consoles

Refers to prefabs	
HSK Base	B2_Cons_Mode, B2_Cons_Power *Assets > Heavy Station Kit > BASE > Prefabs > Equipment*

There are two types of Console prefabs:

Prefab B2_Cons_Power – "the Power console" allows for choosing if the door/gate is either operating properly or inactive;

Prefab B2_Cons_Mode – "the Mode console" allows for choosing if the door/gate is either Active or Blocked.

TIPS

- Consoles can manage all types of HSK Base, Colony, Dugout and Hangars Doors and HSK Hangars Gate2 (**H2_Gate2**) Prefabs simultaneously.
- Both consoles **B2_Cons_Power** and **B2_Cons_Mode** aren't available for manipulation if the first door in theirs ***ControlledDoors*** list has mode either ***brokenOpen*** or ***brokenClosed***.
- The Console **B2_Cons_Mode** does not work if the first door in the ***ControlledDoors*** list has mode either ***inactiveOpen*** or ***inactiveClosed***.

SETTING UP THE CONSOLE

1. Attach the script ***DotHskDoorControl*** (*Assets > Heavy Station Kit > _common > Scripts > Doors > DotHskDoorControl.cs*) to all instances of the door prefab, which you would like to manipulate.

2. Set ***DotHskDoorControl*** script parameters:

2.1. ***OpenIfPowerOff*** to ***true*** for the door that you would like automatically opened if the power will go down.

3. Specify ***PowerOnStatus*** so after the Power is restored doors will get:

blocked	the doors will get locked, and the Red light will signalize that
active	the doors will get unlocked, and the Green light will show this
previous	the doors will get into their previous state when the Power went off. If initially inactive, then the value set at ***BlockedByDefault*** parameter will be used

4. Specify all the doors/gates to be controlled via this particular console, using the ***ControlledDoors*** parameter (of the **DotHskDoorConsole** script, which is attached at the instance of the console prefab). The same doors/gates can be placed to ***ControlledDoors*** list of many consoles.

5. Check the ***ConsoleList*** parameter of the **DotHskDoorControl** script, for there should be all the consoles that are controlling this door. Please do not edit this list, because it is automatically made.

TIPS

To set a mode for multiple doors which are handled by single console, specify the mode of the first door in the ***ControlledDoors*** list. If necessary, multiple consoles can manage one door and a single console can manage many doors. If having such a tricky situation, please keep in mind:

- the mode of the first door in the ***ControlledDoors*** list is displayed by the console, and only the mode of the first door in that list is taken into account when switching modes;
- all the consoles that handle the same doors are equal in functionality.

Be careful at making complex door control configurations. If set up incorrectly, some doors may get into unexpected modes.

If the doors are operated by console, it is recommended to switch their mode using the following methods of the **DotHskDoorControl** script attached to the first door/gate object in the ***ControlledDoors*** list:

*Useful public methods of **DotHskDoorControl** class*	
void **SetPowerMode(** *bool* ***isOn*** **)**	Allows to switch on/off the Power of the door. For each door, this method saves and restores its stance "active/blocked" and considers the value of the parameter ***OpenIfPowerOff***. Acceptable values for ***isOn*** parameter are bool ***true*** (for turning the power on) or bool ***false*** (for turning the power off).
void **SetMode(** *dotHskDoorMode* ***mode*** **)**	Allows doors/gate blocking and unblocking. The method can set off any of the available modes; however, for switching the power on/off, it is recommended using **SetPowerMode()** method. Acceptable values are ***dotHskDoorMode.{mode_id},*** where ***mode_id*** is one of following literals: ***active***, ***blocked***, ***inactiveOpen***, ***inactiveClosed***, ***brokenOpen***, ***brokenClosed*** (see list of operating modes above).
DotHskDoorConsoleCollider *script (attached to Console_Trigger GameObject)*	
Texture **Banner**	On-screen hint image (source file **HSK_Gui.psd** included in *Assets > Heavy Station Kit > _common > Textures > GUI*)

Elevator

Refers to prefabs	
HSK Colony	C_El_Platform, C_El_Platform2 *Assets > Heavy Station Kit > COLONY > Prefabs > Elevator*

SETTING UP THE ELEVATOR

1'st Step. Place the Platform (Cabin) of the Elevator in the scene.

TIPS

Two platform types are available and they differ by pre-installed consoles:

- *The platform **C_EL_Platform** is using console **C_EL_Console**, which provides keyboard input for selecting specific floor, and for selecting underground level stories an additional modifying button should be used.*
- *The platform **E_EL_Platform2** is using console **C2_El_Cons2**, which shows list of the available floors on the graphical panel, and allows selecting of the required floor using mouse button via "touchscreen".*
- ***Tip**: While operating touchscreen elevator console "C2_EL_Cons2" Player may have an item in their hands.* Usually, the *use of console behaves through pressing the same button, which is binded for use of an item in the hands of the Player - i f that is the case, you can add callback-functions (see the "Recommended **Specific solutions**" Easy FPS section below)*

2'nd Step. Place Consoles of the Elevator on all floors and at the Platform of the Elevator.

TIPS

Coordinate at Y axis of the Console's origin point is used for positioning Platform of the Elevator on according floor.

3'rd Step. Script setup

A. Main settings (script DotHskElevator2, assigned as child component to Platform Object):

1) Optional, only for custom (non **C_El_Platform** or **C_El_Platform2**) platforms:

- Assign to property "Platform" - Platform object.
- Attach Platform Console:
  - for **C_El_Console** - assign to property "Platform Console" of DotHskElevator2 script Console object that is placed at the Platform of the Elevator.
  - for **C2_El_Cons2** - assign Platform object (**C_El_platform2**) to property "Elevator 2" of DotFPCElevator2ConControl script attached to **C2_El_Cons2** prefab.

2) Set number of Floors of Elevator at property "Size" of list "Floors" and to each element of the list:

- Assign appropriate Console objects to property "Console"
- At "Elevator Label" property set a symbolic ID code of Elevator title for displaying on digital panel (only for **C_El_Platform2**)

TIPS

At Console assignment, readonly "Floor Height" property of an appropriate element of list "Floors" shows the height of the floor (Y-axis).

- Using slider bar "Floor Number" for each element set hotkey for selecting Floor number at Console of Elevator.

TIPS

At configuring script while in Edit Mode, numbers of floors are automatically modified, so they stay unique;

Supported range of the number of floors is from "-9" to "9". While in game, hold modifier key (by default "Shift", can be changed via DotControlCenter prefab) to type in Negative, or in other words, Underground floor number.

- At the "Floor title" property set floor title for displaying on a digital panel (only for **C_El_Platform2**).

3) At the "Current floor" property set floor, on which Platform of Elevator will be at start of the game. So the platform should move to such floor.

TIPS

At this property should be assigned an index of the appropriate element from the "Floors" list. This differs from the actual floor number.

4) Set platform movement speed at the property "Platform Speed".

B. Optional - customize movement sounds (script DotHskElevator2, assigned as child component to Platform object of the Elevator):

1) Assign to property "Platform Sound Source" AudioSource object, attached at Platform of the Elevator.

2) Assign at "Start Sound", "Motion Sound" and "Stop Sound" properties AudioClip with corresponding sounds, such as starting, movement and stop.

TIPS

Duration of AudioClip "Start Sound" defines the amount of time that takes Elevator to accelerate, and "Stop Sound" - braking of Elevator till stopping.

C. Optional - customize Displays of Consoles (script DotHskElevator2Events, assigned as child component to Platform object of the Elevator)

1) Assign to property "Display Up Mat" material for the upper display of the console, which shows the number of the current floor at standby mode as well as at movement of the Platform.

2) Assign to property "Display Dn Mat" material for the bottom display of the Console at Floors, which shows the state of the Elevator - "Movement up", "Movement down" or "StandbY'.

3) Assign to property "Display Dn Platform Mat" material for the bottom display of the Console at Platform, which shows the number of the desired floor while Elevator is running.

TIPS

*Every elevator that is placed within the scene, **must use a separate set of materials for displays**. Because for showing identical information at Consoles script is modifying property sharedMaterial of Renderer object.* Detailed information on preparing materials for Displays and configuring the **DotAnimatedTexture** script can be found in the "Displays" section below.

*Useful public properties & methods of **DotHskElevator2** class*	
int **currentFloor**	The Property contains the internal number of the current floor, to move the elevator platform use the method **call**()
bool **call**(*int **floor***)	"Call" elevator platform to specified ***floor***, the method will return ***false*** if action can't perform
Texture **CallElevatorTip**, **EnterFloorTip**	On-screen hint images (source file **HSK_Gui.psd** included in *Assets > Heavy Station Kit > _common > Textures > GUI*)

D. Optional – attach callback routines

Attach callback routines for events, arising when graphical panel (*console **C2_El_Cons2***) activated when the player approaches the console and deactivated when the player moves away from the console (script **DotHskElevator2ConControlCol**, assigned as child component to **C_El_Collider** - child of graphical panel console **C2_El_Cons2**) , see example in chapter "Third-party Character Controller Requirements" below

*Useful public properties & methods of **DotHskElevator2ConControlCol** class*	
UnityEvent **OnDisplayActivated**	The property contains callback procedures that are called when the graphic panel is activated*
UnityEvent **OnDisplayDeactivated**	The property contains callback procedures that are called when the graphic panel is deactivated*

** Create callback procedures as script methods attached to some **GameObject** in the scene and assign them to the specified properties*

Shuttle

HSK Dugout kit contains a set of prefabs that implement the functionality of character-controlled shuttles of various designs, including:

Refers to prefabs (HSK Dugout kit only)	
Shuttle *Assets > Heavy Station Kit > DUGOUT > Blueprints > Shuttle Systems*	Shuttle frame *Base shuttle prefab - contains the shuttle frame with connected main control scripts*
DBPS_II_Cons_0, DBPS_II_Cons_1, DBPS_II_Cons_2, DBPS_II_Cons_3 *Assets > Heavy Station Kit > DUGOUT > Blueprints > Shuttle Systems*	Console *Control console for activating the shuttle control mode*
DBPS_EE_Turb_L, DBPS_EE_Turb_Left, DBPS_EE_Turb_Left1, DBPS_EE_Turb_Right, DBPS_EE_Turb_Right1, DBPS_EE_Turb_S *Assets > Heavy Station Kit > DUGOUT > Blueprints > Shuttle Systems*	Turbine *Several variants of animated jet turbines working in conjunction with the shuttle controller*
DBPS_EE_Chas_0, DBPS_EE_Chas_1, DBPS_EE_Chas_2 *Assets > Heavy Station Kit > DUGOUT > Blueprints > Shuttle Systems*	Chassis *Several variants of animated chassis*
Follow Camera *Assets > Heavy Station Kit > DUGOUT > Blueprints > Shuttle Systems*	Follow Camera *Follow camera for organizing the user interface when controlling the shuttle*

SETTING UP THE SHUTTLE

1'st Step. Place the prefab **Shuttle** in the scene and build the shuttle body from the components of the DUGOUT and other HSK packages, place the interior elements, install the landing chassis, turbines and control consoles, adhering to the following rules:

1) all elements of the shuttle structure, including turbines, chassis and static cameras, should be placed in the **Shuttle Model** container, and it is desirable to place static cameras in the **Static Cameras** folder.

2) all **Follow Cameras** must be placed outside the **Shuttle Model** container, it is recommended to place them in the **Follow Cameras** container.

3) If it is necessary to organize remote control of an unmanned shuttle from the ground, the corresponding control console should be located outside the **Shuttle Model** container.

2'nd Step. Add to the **Flight Colliders** object the minimum number of **Sphere**, **Capsule** or **Box** Colliders needed to roughly represent the fuselage shape. These colliders are necessary for physical interaction with other objects in the scene at the time of flight, since at this time all other colliders attached to **Objects** in the **Shuttle** container are disabled.

3'rd Step. Script settings

1) DotHskShuttleSupports (attached to **Shuttle Model** object in **Shuttle** prefab)

- Attach first person controller (**FPC_Player** prefab) to **Person Controller** property
- Attach shuttle turbines to array **Turbines** property

- Attach shuttle chassis to array **Chassis** property
- Attach all (static and follow) cameras to array property **Cameras:**
  - to item **Camera** attach object with **Camera** component
  - to item **Listener** attach object with **AudioListener** component
  - in the **Hot Key** item set a hotkey that will activate the corresponding camera

2) DotHskShuttleFollowCamera (attached to **Follow Camera** prefab)

- Attach **Shuttle Model** object (**Shuttle** prefab) to property **Target**

3) DotHskShuttleTurbine (attached to turbine prefab)

- Adjust (if necessary) the **Place** and **Location** properties, see the **DotHskShuttleTurbine** script below for details.

4) DotHskShuttleCollider (attached to **Trigger** object of **DBPS_II_Cons_{N}** prefab)

- Attach topmost **Shuttle** container object (with **DotHskShuttleController** script attached) to **Shuttle Controller** property.

5) FPC_Shuttle (*Assets > Heavy Station Kit > _common > Scripts > FPC > FPC_Shuttle.cs*)

- Attach **FPC_Shuttle** script to **FPC_Player** object.

Shuttle scripts overview

A) **DotHskShuttleController** class provides the main shuttle functionality - handling user input and flight control

DotHskShuttleController class	
General settings Section	
Rigidbody **Model**	Reference to **Shuttle Model** object with **Rigidbody** attached
Hot Keys Section	
KeyCode **SwitchEngine** (Z)	Hot key to turn on / off jet turbines
KeyCode **Quit** (X)	Hot key to exit flight control mode
KeyCode **Forward** (W)	Hot key for moving forward
KeyCode **TurnLeft** (A)	Hot key for turning left
KeyCode **Backward** (S)	Hot key for backward movement
KeyCode **TurnRight** (D)	Hot key for turning right
KeyCode **StrafeLeft** (Q)	Hot key to strafe to the left
KeyCode **StrafeRight** (E)	Hot key to strafe to the right
KeyCode **Upward** (Space)	Hot key for takeoff up
KeyCode **Downward** (Left Ctrl)	Hot key hotkey for going down
float **TurnForce**	The force applied to the model when turning
Forces Section	
float **ForwardForce**	The force applied to the model when moving forward (backward)
float **ForwardTiltForce**	Force of downward tilt of the shuttle bow when moving forward
float **TurnTiltForce**	Lateral tilt force of the shuttle when turning and / or strafe
float **StrafeForce**	The force applied to the model when strafe
float **LiftForce**	Force applied to the model when going up and / or going down
float **FreeFallForce**	The force of "gravity" applied to the model in free fall when the engines are turned off
float **TurnTiltForcePercent**	Percentage of tilt power when turning
Wiggling Section	
float **WiggleAmplitude**	The angular amplitude of the shuttle tilts when wiggling. The shuttle tilts between 50% and 100% of the specified

	amplitude
float **WiggleDuration**	Oscillation period of wiggling, sec
float **WiggleDelay**	Delay from the last press of any control key until the shuttle enters wiggle mode, sec
float **WiggleMinHeight**	The minimum height of the shuttle hovering above the surface, starting from which the wiggling mode can be activated
	Miscellaneous Section
float **FreeFallHeight**	The height of the shuttle above the surface during downward movement, below which the descent speed begins to be limited. This prevents the shuttle structural elements from falling under the surface upon landing
	Other public properties and methods
bool **Operate**	R/W Property. Activates (true) or deactivates (false) shuttle control mode. When the shuttle control mode is activated, the character controller is disabled and the shuttle control interface is activated
bool **EngineAct**	R/o property. Contains "true" if engines are activated and "false" otherwise
bool **OnGround**	R/o property. Contains "true" if the shuttle is on the surface (on ground) and "false" otherwise
OnChangeStatus **changeStatus**	R/W Property. Callback (delegate) that is activated when the shuttle status changes, declared as *public delegate void OnChangeStatus(Rigidbody rb, HSKShuttleStatus param, Vector4 control), where* • ***rb*** – *Ridgitbody* of shuttle • ***param*** – type of event (*enum HSKShuttleStatus):* ○ ***stNone*** – no events ○ ***stBegin*** – activate shuttle control interface ○ ***stEnd*** – deactivate shuttle control interface ○ ***stStart*** – start engine ○ ***stStop*** – stop engine ○ ***stControl*** – user input received • ***control*** – array (Vector4) of user input: ○ **item 0 / x** – Turn Left / Right ○ **item 1 / y** – Forward / Backward ○ **item 2 / z** – Upward / Downward ○ **item 3 / w** – Strafe Left / Strafe Right

B) **DotHskShuttleCollider** class provides operation of shuttle console (activation of the shuttle control mode) and displaying the GUI prompts

DotHskShuttleCollider *class*	
DotHskShuttleController **ShuttleController**	Reference to the topmost shuttle container object with an attached **DotHskShuttleController**
KeyCode **Interact**	Hotkey to activate the shuttle control mode
Texture2D **shuttleControlStartTip, shuttleEngineOnTip, shuttleFlightModeTip**	Images for corresponded GUI prompts
bool **DisplayGUIMenu**	If true, then GUI prompts are displayed, does not apply to the **shuttleControlStartTip** prompt, which is displayed when the character interacts with the control console

C) **DotHskShuttleSupports** class, provides control of turbines, chassis and cameras, and is also responsible for interacting with the character controller

DotHskShuttleSupports *class*	
DotHskShuttleController **ShuttleController**	Reference to **Shuttle** container object with **DotHskShuttleController** script attached
GameObject **PersonController**	Reference to **FPC_Player** object (first person controller)
KeyCode **ToggleChasis** (C)	Hotkey for opening / closing the chassis
List<DotHskShuttleTurbine> **Turbines**	List of references to **Turbine** objects with **DotHskShuttleTurbine** script attached
List<DotHskMov> **Chasis**	List of references to **Chassis** container object with **DotHskMov** script attached
List<DotHskShuttleCamera> **Cameras**	List of references to **Camera** game object with **Camera** component attached

D) **DotHskShuttleTurbine** class provides control over the audio-visual effects of the turbine operation – rotation, humming and flame display

DotHskShuttleTurbine class	
Transform **Item**	Reference to the **Transform** component of the rotating part of a jet turbine
Renderer **NozzleRenderer**	Reference to game object with **Renderer** of turbine Nozzle attached
AudioSource **SoundSource**	Reference to game object with attached Turbine **AudioSource**
AudioClip **StartSound, MoveSound, StopSound**	Audio clips for, respectively, starting, working and stopping noise f the turbine
Light **NozzleLight**	Reference to Nozzle light source
DotHskTurbinePlace **Place** DotHskTurbineLocation **Location**	Place of the turbine on the shuttle fuselage, affects the tilt of the turbine in the process of changing the direction of flight, allowable values: **Place –** *static, left, right* **Location –** *middle, front, rear* Turbine with **Place** set to "static" remains stationary. In general, the **Place** value affects all types of movement, and **Location** value affects the strafe
bool **PlaySounds**	If **true**, turbine noise will be reproduced
float **BackwardAngle, ForwardAngle, VerticalAngle, IddleAngle**	Boundary rotations of the turbine in appropriate situations
float **Responsivity**	Turbine rotation speed

Note. The source file **HSK_DUGOUT_FONT.psd** with sample font for the shuttle displays and source file **HSK_Shuttle_GUI.psd** with on-screen hints with shuttle modes are included in the *Assets > Heavy Station Kit > _common > Textures> GUI* folder.

Ladder

*SETTING UP THE LADDER**

1. Attach control script (**FPC_Ladder** class) to **FPC_Player** GameObject.

2. Assign tag **Ladder2** to all GameObjects that have Ladder Colliders attached to.

Refers to prefabs	
HSK Base	Ladder1, Ladder2 *Assets > Heavy Station Kit > BASE > Prefabs > Ladders*
HSK Hangars	GW_Ladder *Assets > Heavy Station Kit > HANGARS > Prefabs > Gateway*
HSK Colony	C_Basin_Ladder *Assets > Heavy Station Kit > COLONY > Prefabs > Floors* C_Vent_4m, C_Vent_9m *Assets > Heavy Station Kit > COLONY > Prefabs > Ventilation*
HSK Dugout	DAI_EE_Ladder_0, DAI_EE_Ladder_1 *Assets > Heavy Station Kit > DUGOUT > Prefabs > Type A Interior* DBCP_II_Ladder *Assets > Heavy Station Kit > DUGOUT > Prefabs > Type BC Props* DSP_II_Ladder *Assets > Heavy Station Kit > DUGOUT > Prefabs > Type S Props*

Useful public properties of FPC_Ladder class	
Texture **tipOnLadder, tipOffLadder**	On-screen hint images (source file **HSK_Gui.psd** included in *Assets > Heavy Station Kit > _common > Textures > GUI*)

Settings mentioned above allow for a link between Ladders and class **FPC, which are part of **Heavy Station Kit Asset**.*

Other animated prefabs

Refers to prefabs		
	HSK Dugout kit	
DAI_II_Stairs_Uni	Swivel gangway	*Assets > Heavy Station Kit > DUGOUT > Prefabs > Type A Interior*
DAI_II_End_Vent_Door, DAI_II_End_VentS_Dn, DAI_II_End_VentS_Up	Ventilation grill	
DBI_II_End_Vent	Ventilation grill	*Assets > Heavy Station Kit > DUGOUT> Prefabs > type B Interior*
DCI_II_End_Vent	Ventilation grill	*Assets > Heavy Station Kit > DUGOUT> Prefabs > type C Interior*
DAP_EE_Radar	Rotating radar	*Assets > Heavy Station Kit > DUGOUT > Prefabs > Type A Props*
DBCI_II_Hatch_HSK	Sliding hatch	*Assets > Heavy Station Kit > DUGOUT > Prefabs > Type BC Interior*
DBCP_II_Fan	Industrial fan	*Assets > Heavy Station Kit > DUGOUT > Prefabs > Type BC Props*
DBCP_II_Switch1, DBCP_II_Switch2	Switches	
DS_Door	Doors with folding ladder	*Assets > Heavy Station Kit > DUGOUT > Prefabs > Type S Props*
DSP_II_Capsule_1, DSP_II_Door_Left , DSP_II_Door_Right, DSP_II_Storage_x, DSP_II_WC, DSP_II_WC&Shower etc	Animated furniture and plumbing fixtures	
DBPS_EE_Chas_0 .. 2	Shuttle Chassis	*Assets > Heavy Station Kit > DUGOUT > Blueprints > Shuttle Systems*
	HSK Colony kit	
C_Vent_Grid	Ventilation grill	*Assets > Heavy Station Kit > COLONY > Prefabs > Ventilation*

The prefabs listed above are controlled by the universal **DotHskMov** script

*Public properties of **DotHskShuttleSupports** class*	
dotHskDoorMode **mode**	Operating mode (see list of operating modes in "Doors & Gate2" section)
int **repeatMotion**	If **0 –** movements are automatically repeated in an endless loop, if **1** or more – the movements are initiated by script and repeated the specified number of times
bool **reverseOddCycles**	If **true –** movement performed in the opposite direction in every odd cycle
float **delay**	Pause between movement cycles, sec
float **motionTime**	Duration of one cycle of movement (excluding the pause)
List<dotHskMovFlap> **movFlaps**	List of animated object elements
float **almostOpenPosition**	Flaps position in "Broken open" state (0 – fully closed, 1 – fully open)
float **almostClosedPosition**	Flaps position in "Broken closed" state (0 – fully closed, 1 – fully open)

A helper class that implements the functionality of a separate moving element

*Public properties of **dotHskMovFlap** class*	
Transform **flap**	Refer to Transform component of Flap object
Vector3 **openPosition, openRotation**	Position and rotation of Flap in “Open” state

Vector3 **closedPosition, closedRotation**	Position and rotation of Flap in “Close” state
Int **turnStep**	Angle of rotation for one phase of rotation. If the specified angle is less than 10 degrees, the number of rotation phases is calculated automatically. Phase breakdown of the rotation cycle is performed to ensure smooth rotation.

First Person Character Controller

Refers to prefabs	
ALL Kits	FPC *Assets > Heavy Station Kit > _common > Prefabs*

FPC class is a simple First Person Character Controller class. **FPC** class is independent from deprecated Unity Standard Assets and provides support for all the original functionality implemented in the **Heavy Station Kit** assets.

*Useful public properties of **FPC** class*	
	“Speed” section
float **walkingSpeed**	Regular rate of horizontal movement
float **runningSpeed**	Increased rate of horizontal movement (hold “Shift” by default); has impact on crouch movements
float **climbingSpeed**	Ladder movement rate
float **jumpSpeed**	Jumping rate of movement; has impact on crouch movements
	“Look” section
float **lookSpeed**	Rate of Camera rotation
float **lookXLimit**	Scope of Camera rotation
	“Features” section
float **crouchSpeedRatio**	relation of crouching speed to regular
float **crouchHeightRatio**	relation of crouched Character Controller height to regular
	“Climbing” section
bool **climbingAutoStart**	If True, allows to climb once in Ladder Collider area; else interaction starts by a key (“E” by default)
	Other Settings
bool **canMove**	If False (default True), Player can only sit, look and to be subjected under the effects of gravity.
float **gravity**	Gravitational Force
Camera **playerCamera**	Player Camera

Dot Control Center

Refers to prefabs	
ALL Kits	DotControlCenter *Assets > Heavy Station Kit > _common > Prefabs*

The *DotControlCenter* prefab is made to be a convenient center for centralized control of the general settings of other prefabs from **Heavy Station Kit** set either in an active scene or in the entire application.

Place *DotControlCenter* prefabs in every scene, if it is necessary to control settings of the prefabs individually. On the other hand, to control settings in the entire application place *DotControlCenter* prefab in the starting scene of the project and tick checkbox “Use in Other Scenes”.

List of parameters available for setup:

Shortcuts	
InteractShortcut*	“One-buttoned” interaction (default - "E" key)
CrouchShortcut	Toogling character mode to Crouch/Walk (default - "C" key), applied for *FPC* prefab
FlashlightShortcut	Turning flashlight either ON or OFF (default - "L" key), applied for *FPC* prefab

Shortcut modifiers	
Basement Floors ModifierKey 1, Basement Floors ModifierKey 2	Modifier buttons to input negative floor number on floor selection console of the Elevator, and is used in conjunction with keys "1"..."9" to form an appropriate negative variant "-1"..."-9" (default -"Left shift" and "Right shift" keys), applied for C_EL_Platform
Settings	
Use In Other Scenes	If the check mark is set, the **DotControlCenter** object will not be destroyed when a new scene is loaded
Track Changes Settings	If the check mark is set, the settings changes will be tracked and applied in each update application cycle

*) With the modification of *InteractShortcut* it would be obvious to update corresponding on-screen hints. Graphical source file **HSK_Gui.psd** included in *Assets > Heavy Station Kit > _common > Textures > GUI*

Displays

Refers to prefabs	
ALL Kits	*Assets > Heavy Station Kit > BASE / HANGARS / COLONY / DUGOUT > Prefabs > Displays*

Displays use **DotAnimatedTexture** Script, designed for cycled playback of single or multiple frame sequences assigned to the material.

Prepare material

Recommendations for material creating and setup:

1. Frame sequences must be inside textures. And then you put the textures in the **Main Maps** section of the Material.

2. The size of the texture should provide optimal space for all of the frame sequences.

3. Positioning of frame sequences on the texture map is done in the following order - from left to right and from top to bottom. So at first, the row is being made, then other rows add, filling texture map to the bottom.

4. Setup for parameter **Tiling** for **Main Maps:**

X = 1.0f / {columns_count}, where {columns_count} is a number of frames that are placed horizontally;

Y = 1.0f / {rows_count}, where {rows_count} is a number of frames that are placed vertically.

SETTING UP THE SCRIPT

1. After the material was assembled, assign it to the desired object.

2. To the same object, script **DotAnimatedTexture** is being attached. Script's parameter **Material Total Frames** is set automatically for a maximum number of frames that can be in the material. The chosen number of frames depends on the values of the **Tiling** parameter for **Main Maps**.

3. General script configuring:

Active Sequence – the number of current sequence for playback (zero-based);

Size in **Sequences** tab – total number of frame sequences in the animated material;

FPS – number of frames per second on playback;

Show warnings – allows for displaying errors in Console if configuring the script in EditMode (Disabled by default)

4. Individual setup of single or multiple frame sequences, on tab "Element N" of the "Sequences" tab.

Total Frames – total number of frames of this particular frame sequence;

First Frame – first frame number, of this particular frame sequence element, in relation to the first frame number on the Material (zero-based);

Starting Frame – sequence playback starts with this frame (zero-based);

Randomly — if checked, frames to playback will be chosen on random.

Notes

1. At script setup in EditMode, animated material shows the starting frame (parameter **Starting Frame**) of selected sequence (parameter **Active Sequence**). This allows for a visual preview of animated material. Frame sequence cannot be run in EditMode, for this select GameMode.

2. For switching between frame sequences inside of Script in GameMode it is necessary to assign sequence number (zero-based) to public property **activeSequence** of an appropriate DotAnimatedTexture script component.

3. Please keep in mind that, if making a Prefab from an Object with the already attached script, then assigned material will drop out of Prefab. Restoring material is possible within the Inspector, selecting Prefab in Project window and assigning material manually. Then, for preview picture of Prefab to display correctly, it is advised performing Reimport.

Recommended

General settings for 3rd party FPC

to work with Heavy Station Kit

*(For **Opsive UFPS 2.0** and **Easy FPS** see the **Specific solutions** below)*

1. Character collider should not exceed **1.8m** in height and **0.7m** in diameter.

2. For **most scripts** to respond (interaction with the elements of Heavy Station Kit Asset like Doors, Consoles, etc), tag **Player** must be set either in Character Controller collider or in any of its parent Game Objects.

3. Elevator display console C2_El_Cons2 (included in Heavy Station Kit colony asset) requires tag **MainCamera** set in player camera.

4. To use **Ventilation** 3rd party FPC must support **crawling** or **crouching**. Additional requirements are height of player collider **<0.8m**, height of player camera **<0.65**. Size of vent unit is **1m x 1m**.

5. To use **Ladders** (including ventilation ladders) 3rd party FPC must support that.

6. To avoid falling through in narrow space parameter **Clipping plane: Near** must be set at lowest point **0.01m**.

Recommended

Specific solutions

*for Heavy Station Kit to work with **Opsive UFPS 2.0** and **Easy FPS***

Opsive UFPS 2.0

https://assetstore.unity.com/packages/templates/systems/ufps-ultimate-fps-106748

Issue: All UI elements of UFPS got added to console display C2_El_Cons2 instead of main screen, after adding UFPS UI to scene which has console display.

Solution: It happens because C2_El_Cons2 has CANVAS and UFPS2 script applies UI to first CANVAS it can find. Simply **temporarily disable** all Elevators which has C2_El_Cons2 *mounted on E_EL_Platform2* **before** adding UI *(Tools→ Opsive → Ultimate Character Controller → Main Manager → Setup → UI Setup → Add UI).* Don't forget to **enable all Elevators back** when UI was added.

TIP: Out of the box UFPS doesn't support **Crawling** (*Ventilation*) and **Climbing** (*Ladder*). Yet you can use their documentation to implement that features yourself at:

https://opsive.com/support/documentation/ultimate-character-controller/character/abilities/new-ability/

Easy FPS

https://assetstore.unity.com/packages/3d/characters/humanoids/easy-fps-73776

Issue: While operating touchscreen elevator console "C2_EL_Cons2" Player may have an item in their hands. Usually, the use of console behaves through pressing the same button, which is binded for use of an item in the hands of the Player - the possible solution to add a script and to attach methods:

Solution Step 1

Attaching **script**. Simply add script listed below to any GameObject within scene, *(or include the code to existing script)*:

```
using UnityEngine;
class ElevatorDisplayToggle : MonoBehaviour {
   public GameObject playerObject = null;
   private GunInventory playerScript = null;
   private bool currentGunState = true;
   public void Start() {
      if( playerObject != null ) { playerScript = playerObject.GetComponent<GunInventory>(); }
   }
   public void OnDisplayActivated() {
      // Some code that is executed when the character approaches the display console
      if( (playerScript != null) && (playerScript.currentGun != null) ) {
         currentGunState = playerScript.currentGun.activeSelf;
         playerScript.currentGun.SetActive(false);
      }
   }
   public void OnDisplayDeactivated() {
      // Some code used when moving a character away from the display console
      if( (playerScript != null) && (playerScript.currentGun != null) ) {
          playerScript.currentGun.SetActive(currentGunState);
       }
   }
}
```

Solution Step 2

Attach **methods** OnDisplayActivated() / OnDisplayDeactivated() to graphic display console collider (ELEVATOR game object → C_El_Platform2 game object → C_El_Collider game object → DotHskElevator2ConControlCol Script component → On Display Activated () / On Display Deactivated properties)

Installation Guidelines

Step 1. Create clean project (recommended).

Step 2. Install Unity – Post Process Package (recommended).

Step 3. Download and install Heavy Station Kit asset.

Current version of this documentation can be downloaded from
https://dotteam.xyz/pdf/Heavy_Station_Kit_2021.pdf

info@dotteam.xyz
https://dotteam.xyz