R-7 Early Missions

Version 1.0

For Orbiter 2006 P1

Addon for Orbiter Space Flight Simulator
by igel (Andrew Thielmann)

atil@pin-plus.ca

With the exceptional help and inspiration by:

- Thorton (Nikita Vtyurin)
- astronavt (Dmitry Pritykin)
- Bloodest (Roman Vasilyev)
- Russian Orbiter forum community

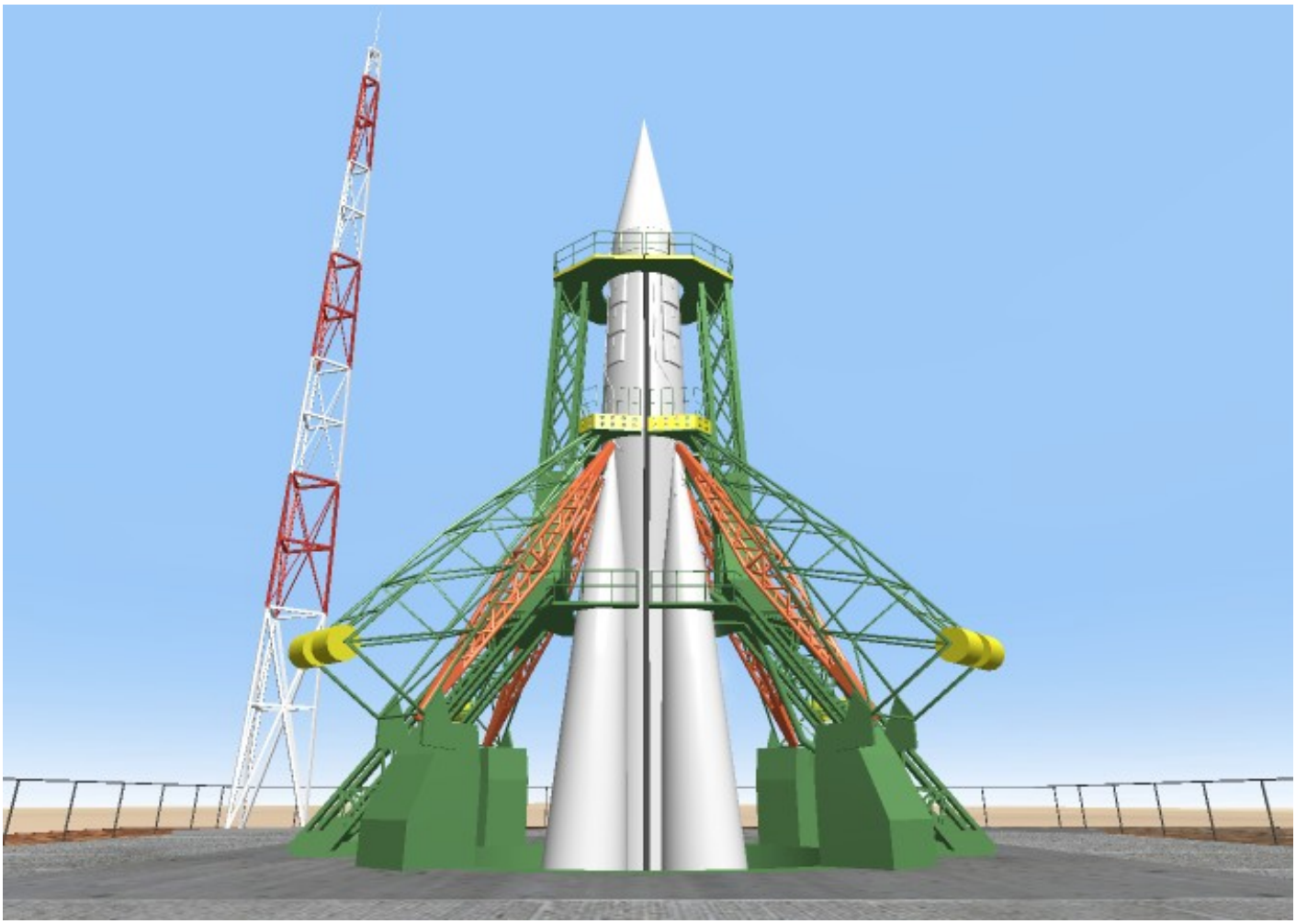


Table of Contents

Quick Start

The easiest way to quickly check out this addon is to run the supplied scenarios. They are grouped into four "R-7.Chapter ..." subfolders under the "Project R-7" folder. All scenarios start and proceed automatically in full autopilot mode. Scenarios and folders descriptions outline the historic background behind each scenario.

Below is the one-page list of all **keyboard commands** (and some important **SCENARIO VARIABLES**) for all vessels of this addon. Print it for quick reference.

All Vessels		**Launchpad**	
Ctrl+F	enable/disable focus for all secondary vessels in scenario	← - →	rotate turntable
TAB	switch focus to the next vessel in the vessel list	↑ - ↓	move service towers up or down
Shift+TAB	switch focus to the previous vessel in the vessel list	↑8 - ↓2	move rocket supports up or down
Ctrl+D	activate self-destruct (if equipped)	**G**	drop rocket supports
C	switch to the next internal camera (if equipped)	←4 - →6	move service platform
RELIABILITY	*Scenario variable*, sets reliability in % (where implemented)	**]} - [{**	move cable mast(s) up or down
		K	drop cable mast(s)
		>. - <,	move fueling mast up or down
		N	drop fueling mast
		M	toggle automation of the launch sequence
		L	launch rocket immediately, bypassing the launch sequence
Block A (core stage)		**Polyot**	
M	toggle between manual and autopilot control mode	**Space**	select next target, if within range
J	jettison (sequentially): • Blocks B-D (strap-ons) • Payload fairing • Payload	**L**	lock on the selected target
Ctrl+J	enable/disable automatic focus switching to payload after its separation	**K**	kill the locked target
Sputnik-1-2-3		**Warhead**	
B	toggle the playback of the radio sound	**NUKE**	*Scenario variable*, equips warhead with nuclear device

Project Scope

R-7 is by far the world's longest-operational rocket. Its "derivatives" are still active and busy nowadays, launching all kinds of space payloads - including the ultimately precious "payload" of human crews. During these long decades, the rocket had nailed more historic "firsts" than one would dare to count!

This long life had brought the rocket itself and its supporting ground infrastructure trough so many modifications and upgrades, that it is not realistic to try to capture them all in single addon. If you try to delve into fine-grain details, the variety and complexity just explode in numbers!

So, the only reasonable approach is to stage the development into phases. This addon is specifically limited to the earliest phase of R-7 life. It covers only the two-stage version of the rocket, and only the historic period between 1957 (when the first test flights of the rocked started to fly) and 1964 (when the last two-stage R-7 was used for space-bound launch).

Other rocket derivatives and other historic periods are the subjects of separate developments. Here, however, limiting the scope to the specific subset of times and configurations has allowed to dive into much deeper details of the early R-7 operations, than it would have been possible otherwise. So, I hope, no one will be disappointed.

Other Recommended Addons and Compatibility

This addon is designed specifically for the Orbiter version 2006 P1 - which is, at the time of release, the most wide-spread and long-stable version of the Orbiter. As an encouraging news, it is being reported to work in most recent Orbiter betas, so this compatibility may happen to extend to future release version (though no dedicated effort was made for that).

Technically, addon has no external dependencies whatsoever, and runs on clean version of the Orbiter "out of the box". However, there are three other addons that are so important for the full and complete experience, that they should be considered as prerequisites (install them **before** this addon):

Project R-7
http://www.orbithangar.com/searchid.php?ID=2844

This is the original R-7 addon by astronavt, released in 2007. It has the amazingly wide and versatile coverage! Even though this new "Early Missions" addon replaces much of its original functionality and some historical missions, the original addon is not looking for retirement yet. It is still required as the basis for many "spin-off" addons, from Lunar missions to Soyuz/ISS operations.

OrbiterSound
http://orbithangar.com/searchid.php?ID=3889

This is a must not only for any dedicated Orbiter fan, but even for the casual tryouts. The new R-7 relies heavily on sound effects to bring realism level up - way up!

Baikonur Surface Tiles v1.0
http://www.orbithangar.com/searchid.php?ID=4247

This addition fills a very wide area with the photorealistic scenery of the Baikonur surroundings. For R-7, it does to visual perception what OrbiterSound does to audio presentation.

Being completely self-contained, the "Early Missions" addon relies on some "tricky" coding in the implementation of its modules, and in the way it uses some of standard Orbiter features. This may lead to compatibility issues with other popular Orbiter addons. Additionally, no special testing was done for any of the regular Orbiter features that are *not* required to run this addon.

History of R-7 Operations

R-7, the first ICBM

R-7 was the world's first truly intercontinental ballistic missile. The test flights of the original version started in May of 1957. The first partially successful flight was conducted in August, and the rocket was announced to the world as operational.

The work, however, was far from over. Warhead was not quite ready for action. The original sharp-nose design could not survive the fiery reentry. Warheads were disintegrating over their target in Kura test range (Kamchatka Peninsula, on Russia's Far East). A new blunt-nose design was introduced to solve the problem. But then another problem became evident: with its 5 tons of gross weight, the warhead could only fly to 8000 km. This was insufficient for military operations...

So the third, and final, modification of the rocket was introduced. Its warhead was weighting only 3 tons. Still, due to the advances in the nuclear technologies, it delivered the same 3 megaton charge. This new rocket was able to fly over 12000 km, and could reach any the intended target in the USA.

This final rocket modification was named R-7A and was put into operational service. Two rockets with live warheads were stored on each of the four launch sites. Four major US cities were selected as potential targets.

This addon comes with the pre-built scenarios, that will allow you to fly-test all three modifications of R-7. If you are going to build your own scenarios, you don't have to specify which modification to use. Addon will set the appropriate version based on the date of the scenario.

R-7A ICBMs stayed quietly in service until 1968. Their most memorable time came in 1962, during the Caribbean Crisis. The rockets were taken from storage and installed on the launch pads - same pads, that, to this time, were mostly used to launch all kinds of space probes and even manned missions! Live warheads were ready to fly off on the shortest notice. In the supplied scenarios, the whole chapter is devoted to what could have happened...

R-7, the first space launcher

4 October, 1957. This date still sends goose bumps down the skin of any space buff. This day, the Space Era has begun, with the launch of the planet's first artificial satellite. There were many launches since, and will be much more in the future, but the first one will forever stay in history...

Due to the initial "overweight" warhead design, R-7 had a lot more power than it really needed for the ballistic flights, and was soon overshadowed in the country's nuclear shield by its slimmer rivals. But this excessive power proved very useful when time came for space launches!

In the short period of late 50-s, R-7 had launched three satellites, one after another, in a quick succession. Each new satellite was adding something new and exiting on top of its predecessors. The whole series of launches was a solid proof of the seriousness of the country's newly born space program. The first simple satellite was not just a "lucky strike" - Russia came to space to stay.

However, the life of the R-7 as a two-stage space rocket was over - just after three launches. New heavier, more ambitious missions required even more power. Very soon, several versions of upper stages were developed for R-7, and the original two-stage version has faded into history.

The new "R-7-derived" rockets added many more spectacular successes to the family's fame. They are not part of this project - because they deserve separate full-scale development streams, and are (or will be) available in the separate addon packs. In this addon, you can enjoy the detailed recreation of the first three historic space missions of the original R-7 rocket in the supplied scenarios.

R-7, the first ASAT tester

Two-stage R-7 made a surprising comeback in the early 60-es. This happened in the course of the development of Russia's first ASAT system. But, by the time of the first test flights, when the first ASAT test items were ready to fly - the rocket, that was supposed to propel them into orbit, was lagging too far behind in its development...

R-7 was called for duty once again. Two two-stage rockets were quickly rigged for orbital trajectory, and launched two test items into space. The missions were officially named Polyot (stands for "Flight" in Russian), and were described as "the first maneuverable satellites". For a very long time they remained the most mysterious Russian space probes - no pictures, or drawings, or technical specifications were ever released until the fall of the Iron Curtain in late 80-es.

Even nowadays, what we know about the Polyot spacecraft is barely enough to recreate it in Orbiter. As it turned out, it was one heck of a space-to-space missile! With the supplied scenarios, you'll be able to fly the two historic Polyot missions. You even have an enhanced mission, that tests the actual ASAT capabilities of Polyot against a material target - something that has never been done in the real life!

Common Features for All Vessels and Scenarios

Historical Authenticity and Immersion Features

For a historical addon, attention to details is crucial. A tiny but characteristic detail can suddenly become a highlight of the user's experience, and make a model truly believable. On the other hand, few things can ruin the immersion effect as destructively as sudden bloopers in a smooth flow of the historical recreation!

For this addon, I used my best knowledge of the R-7 history, with the help of the wide array of historical internet resources. I hope I have introduced sufficient number of the winning mini-details and avoided big loosing bloopers, to provide a pleasant and uninterrupted experience. I also tried to make addon not only accurate and educative, but also to be fun to play with. For this, I looked at other games and simulators for the clues of what makes game fun and provides a smooth and continuous end-to-end experience.

One such "playability" feature" is the presence of the goal in every mission. It is not something that is explicitly set before the mission, or something that ends the simulation once the achieved. The goals come out "naturally", and don't interrupt the missions. You just get a simple message telling you about the success or failure, when it becomes evident.

Special effects are very powerful tools to create immersion feeling, so this addon is full of them. They are not just "effects for effects' sake" - every effect highlights some important operation, change of state or conditions, something that prompts for attention. There are also many sound effects in this addon, that supplement the special effects in storytelling.

Mishaps implementation is another integral part of this addon. The original R-7 was still a learning tool for the young Soviet rocket industry, and the learning curve was steep! Any historical addon would be incomplete without implementing "the dark side" of the story - and this is where the special effects are essential. Guidance failures, engine fires, fuel tank explosions, self-destruct devices, mid-air breakups from excessive loads, reentry disintegration - all this is implemented not just "for the fun of blowing things up". It provides great insights into the issues of rockets reliability, safety problems, debris field evaluations, rocket overfly zones safety, nominal and anomalous stage impact areas, reentry areas and many other aspects of rocketry that engineers always have to deal with.

This addon (just like my previous addons) makes good use of well-thought camera views. Wherever possible, and wherever it makes sense, I try it avoid "generic" empty cockpit view, and set camera (or cameras) to look at something interesting. Sometimes I don't even have to invent the views, and just

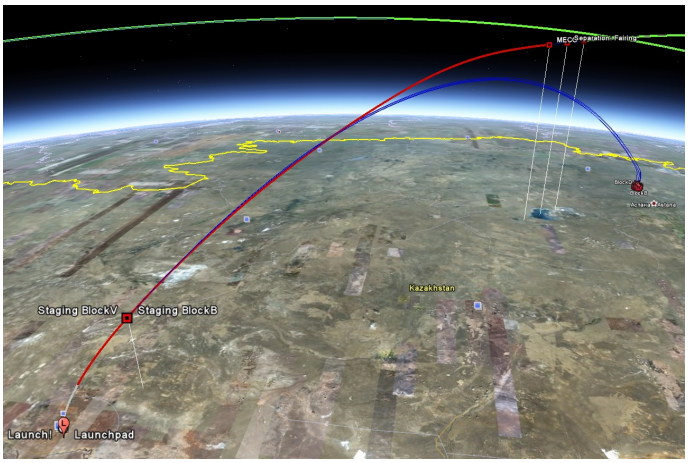
follow the real camera views - because the engineers working on The Real Thing had already taken care of it. When there is more than one camera on the vessel, you can rotate through them by the **C** key (stands for "**C**amera").

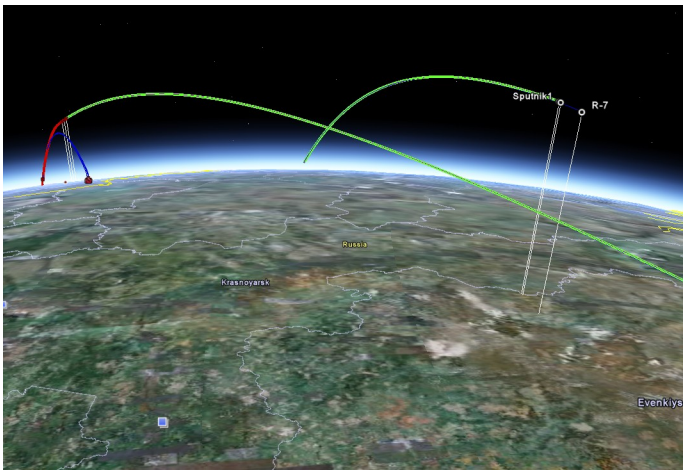
With the advent of Google Earth, nowadays it would be a shame to miss an opportunity for a mashup in such a historically and geographically oriented addon! So, after each flight, successful or not, you can review the mission results in that wonderful tool. Simply exit the Orbiter and load the **(Current state).kml** file, that gets re-created after every R-7 mission in the **GoogleEarth\R-7** directory. You can trace the launch phase, staging events, payload separations, impacts and explosions, debris distributions, active and passive flight trajectories, complete end-to-end ballistic flights or an orbital path (just a couple of initial orbits, as there is a limit to the length of a single track).

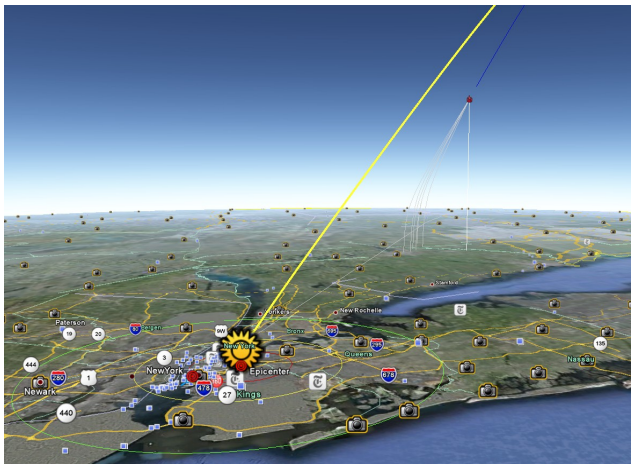
Flight trajectories are color-coded in Google Earth. Active rocket flight is red (becoming thick-white when generating a contrail in the atmosphere). Passive rocket flight is blue. Orbital payload trajectory is green, warhead is yellow, fairing is magenta, and wrecks are thin-white.

For warhead, intended target area circle is plotted, along with the geodetic lines between the launch site, target site and actual impact site (or warhead destruction point). For nuclear warhead, additional total (red), heavy (yellow) and moderate (green) destruction circles are plotted.

Here are a few examples of what you'll see in Google Earth:







Usability Features

After the "playability", the next priority in this addon were the usability enhancements. The two are not the same. Playability maintains fun and interest. Usability keeps things simple (or makes the illusion of the complex things to look simple), shortens the learning period for newbies while retaining full control for power users, hides and automates mundane tasks, guesses or figures out as much needed data as possible from the simulation itself without asking the user, provides convenient shortcuts for frequent operations and does lots of other mini-steps behind the scene. Usability enhancements will make addon more friendly for people outside the regular Orbiter community, who may not be so familiar with the program. But they will be beneficial to everyone using the addon.

The main usability feature is the full automation of all operations. Automation is the primary mode of operations (which is actually historically correct), and manual mode is secondary, reserved for those who know what they are doing. To switch to the manual mode, use **M** key (stands for "go **M**anual").

For manual controls and feature toggles, I try to keep the number of controls to the minimum. And, wherever possible, use convenient mnemonics on the control keys. I also try to provide useful feedback through the onscreen messages, to intercept any "unusual" or "impossible" user input and explain in a meaningful way the reasons behind the interception - and sometimes even to hint how to solve the problem. These messages will, hopefully, reduce the need for this very manual - though reading it is still highly recommended.

I also try to maintain the "human readability" of the supplied and saved scenarios by keeping them "lean". Scenarios only record variables that are different from defaults, which significantly reduces the clutter. Of course, some vessels are more complex than the other, and still need many variables to save state properly, especially in the active phase of the flight.

Orbiter users are accustomed to have control over the pace of time in their universe. They frequently "zoom" through the lengthy sequences at fasts speed, but at the same time don't want to miss anything interesting when it happens. So, this addon is made to be time-acceleration-friendly. It is tested to operate normally in all phases of launch preparation and active flight with time accelerated up to x10, in passive ballistic flight up to x100 and in passive orbital flight up to... well, probably unlimited. Addon has a time-auto-sense feature: when anything interesting is about to happen, that will change the state of the addon significantly - events like MECO, warhead reentry and impact, or engine fire - time speed will automatically return to x1.

There is a similar bypass feature to the lengthy launch sequence. If you are doing many launches when testing some feature, you can shortcut through the launch sequence by the **L** key (stands for "**L**aunch now").

In the addon consisting of so many individual vessels, you may want to frequently switch the focus between the vessels. Unfortunately, focus switching is not a fast and easy feature in Orbiter, so many focus-related helpers are added in this addon to fill the gap.

First comes automatic focus switch through the flight - from launchpad to the rocket, and from rocket to the payload. The first of these two is conveniently delayed by 15 seconds to let you watch all the phases of the launch from the launchpad, and only then switch to the rocket.

Second comes the **TAB** or **Shift-TAB** navigation, that works exactly like switching between application windows in the computer: it switches every time to the next (or previous) vessel in the vessel list, without having to open the list itself. Of course, this feature works only with the vessels provided by this addon. Any other vessels, if added to the scenarios, will be bypassed by the vessel focusing sequence.

Finally, there are simple ways for you to control what can be focused and what not. It is always hard to achieve proper balance on this subject with the rockets consisting of so many big and small parts:

strap-on boosters, fairings, interstages... In most cases you want to disable the focus on the "insignificant" parts, so that the list of "important" focusable vessels is as short as possible. But occasionally you may want to focus such insignificant piece - to get better visuals on separation, to trace its subsequent fate, to see where it falls and if it makes it through the reentry... in this case, you need such piece to be focusable!

This dilemma is solved by the new "focus toggle" feature. By default, the "unimportant" components (and their wrecks) are not focusable, which keep the vessel list tidy and clean. But you can make all vessels in the scenarios focusable by **Ctrl+F** key combination. Pressing the keys once again will revert focus settings for all vessels in scenario to their original, default settings. Of course, this feature also works only with the vessels provided by this addon.

You can also override the default focus behavior of any individual vessel with the **FOCUS** scenario variable (set it to 0 or 1, depending on what behavior you want).

Addon Engineering

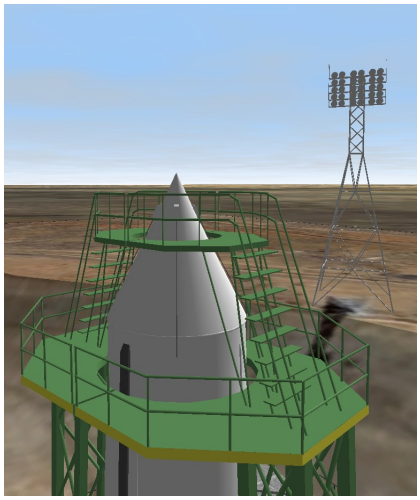
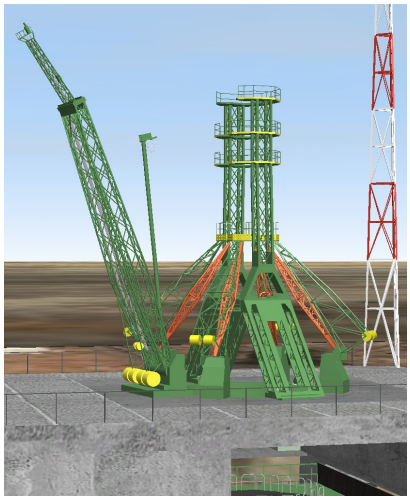
This addon took longer to accomplish and required more work that my previous addons. All those additional features required a new level of integration between modules, more deep and more active than in any of the previous projects. So, extra work was needed to implement this new cross-vessel communications layer.

The layer behaves as the additional API layer on top of the regular VESSEL2 API. It is not a regular universal API, being tied to the particular implementations of this particular addons group. But it has many potentials for future development, enhancement and reuse, and may one day evolve into a completely "addon-agnostic" version.

For curious developers, full source code is provided, with no restrictions.

Individual Vessels

Launchpad Number 1

R-7 launchpad is a facility so complex, so big, so capital and at the same time so "unstable" in time, always evolving, changing its configuration, adjusting to the new types of rockets and payloads, that it will be a task of tremendous complexity to capture all this history in single addon. I am far from thinking that I managed to get it all, nor that I got it all right. But I did my best to get it as close to the historical realities as possible.

I helped myself a little by limiting the scope of this addon. I did not have to account for all varieties of the launchpad configuration through more than a half-a-century of changes. No did I have to trace the different history of all seven launchpads existing today.

In terms of time periods, this addon is limited to the years up to the late 1960-es, and implements configurations for the first test ICBM launches, first satellites launches and later 3-stage Vostok and 4-stage Molniya Rockets. These two last rockets are not part of this addon - but some of the two-stage R-7 flights simulated here were taking place in the early 60-es, when the launchpad was already re-configured for the taller rockets. So you will see this "extra" infrastructure in some scenarios, even though it stays idle and unused in the R-7 launches.

In terms of geography, this addon is limited to only one launchpad, the very first and most historic "Pad Number One". For more authentic representation, it would have been beneficial to implement all six launchpads (two in Baikonur and four in Plesetsk) that were operational during the specified historical period. But this turned out unrealistic to implement in reasonable time, mostly due to significant configurational and landscape differences between pad number one and later pads. Such enhancement is a subject of additional development, and will be added either to subsequent versions of R-7, or to separate addons.

Even in its early configuration, the launchpad of R-7 was so packed with various mechanical systems, that it was a lot of fun to make and, hopefully, will be a lot of fun to play with, even with no rocket installed. This is why the launchpad is implemented as a separate 'vessel', even though it does not fly anywhere. The launchpad hosts a huge turntable (on which most of the other systems are mounted), two big service towers, four rocket supports, varying configurations of cable and fueling masts, and a massive retractable service platform under the table.

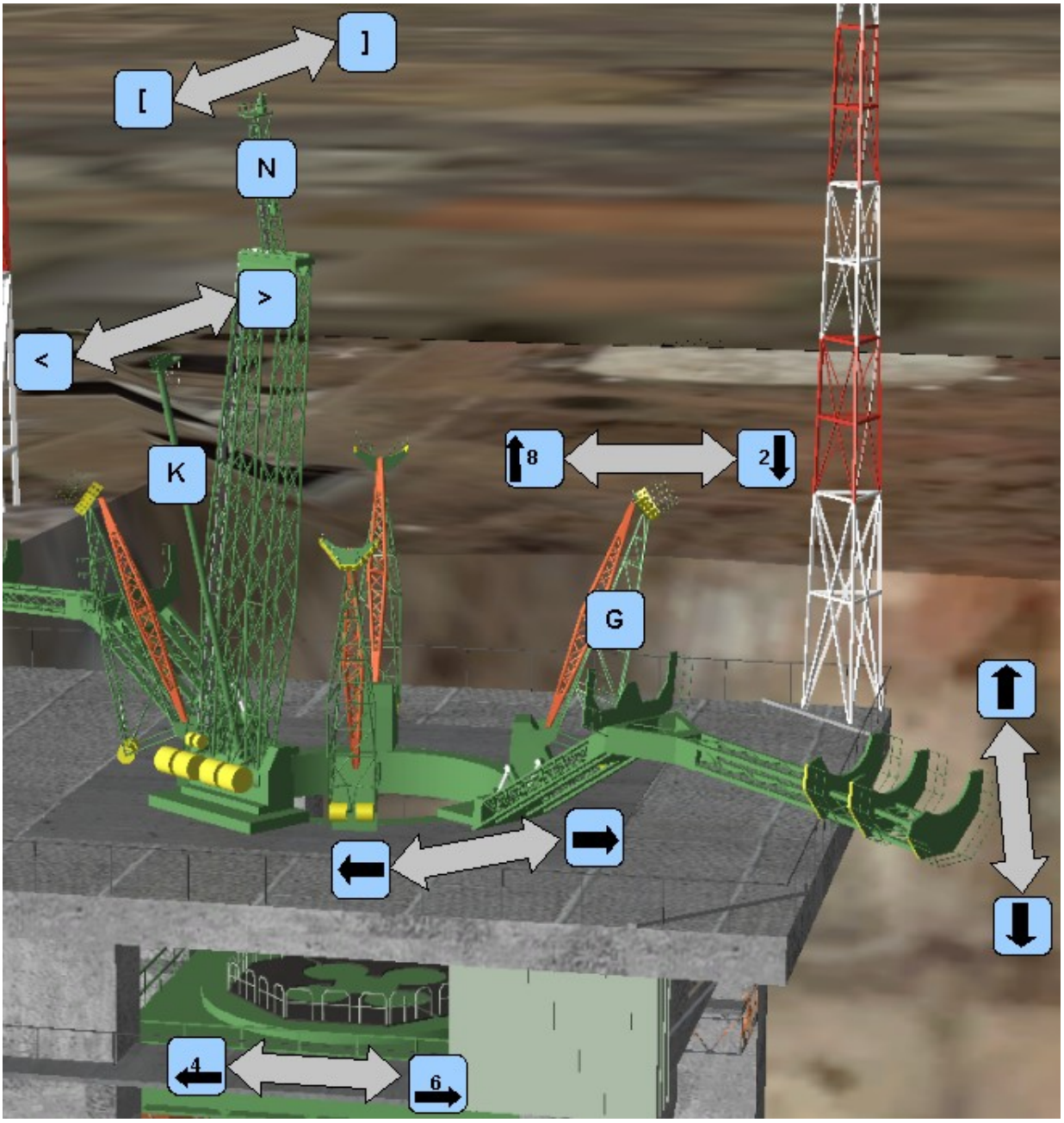
The process of installing the rocket on the pad is not implemented in this addon - yet (but I also hope to add it in future versions). Still it is interesting enough to be familiar with. First, the turntable gets rotated in the position ready to accept the rocket, and all cable masts, service towers and rocket supports are retracted. Then, the rocket is delivered on a special railcar, and is erected vertically, so that its bottom gets lowered pretty deeply into the pit of the table. This reduces the loads from prairie winds (and from nuclear blasts). Then, the rocket supports move into place and "grab" the rocket by its "belt" at the points of strap-on mounts. Finally, the service towers are lifted into the working position, and cable masts are attached to the rocket's umbilicals.

For the launch, the turntable is rotated to a particular orientation, so that the rocket is facing the launch azimuth. Service towers are lowered, and service platform under the engines is retracted into its protective niche. Automated launch sequence is activated. It consists of several important pre-launch steps, including fuel tank pressurization, cable masts drop-off, engines ignition and power-up. When engine's thrust exceeds rocket's weight, the rocket lifts from the supports, which are immediately retracted away from the rocket by heavy counterbalances.

In this addon, all mechanical systems on the pad can be controlled manually. Here is the list of the key controls:

← - **→**	rotate turntable
↑ - **↓**	move service towers up or down
↑8 - **↓2**	move rocket supports up or down
G	drop rocket supports
←4 - **→6**	move service platform
]} - **[{**	move cable mast(s) up or down
K	drop cable mast(s)
>. - **<,**	move fueling mast up or down
N	drop fueling mast

It is not easy to "match" such a long list of controls to the controlled hardware, so a picture might be more helpful:



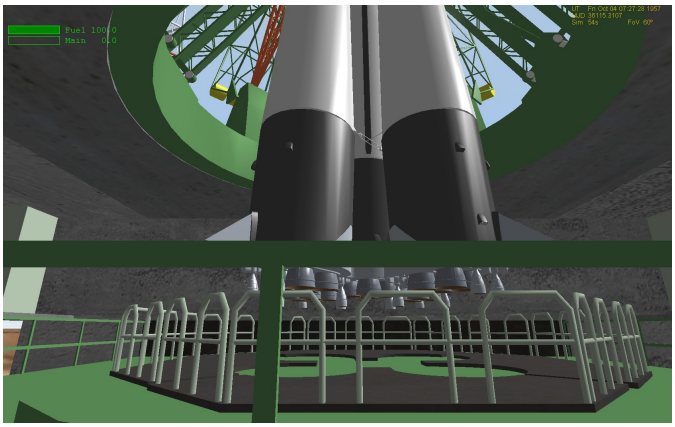
There are some other key controls and scenario variables available in the launchpad:

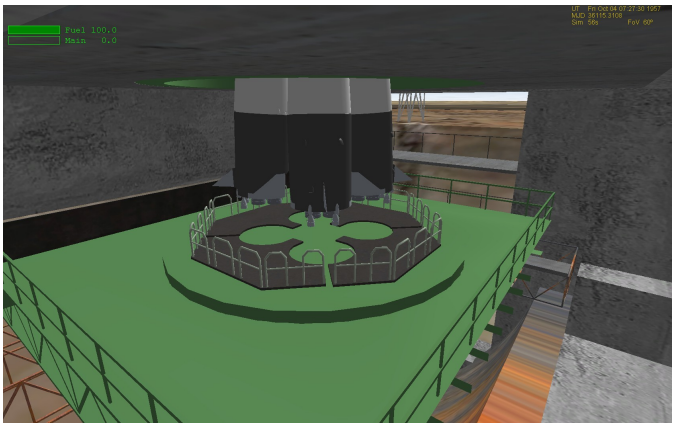
C	switch to the next camera in the "cockpit" (rather, bunker) view
M	toggle automation of the launch sequence
L	launch rocket immediately, bypassing launch sequence
F8	show/hide targeting panel in the "cockpit/bunker" view
LAUNCH_MJD	*scenario variable*, exact planned takeoff moment, in MJD

In the following pages, I'll speak about these functions in better details, one by one.

CCTV "bunker" cameras

As launchpad is "vessel" in this addon, it has "cockpit' view. Which, in the absence of actual "cockpit", becomes 'bunker" view. It has three CCTV camera views (two of which are shown below), strategically placed to provide good views. You can rotate through the camera views with the **C** key (stands for "**C**amera").





Automated launch sequence

Launchpad implements the automated launch sequence, which is "on" by default. It starts 100 seconds prior to the actual takeoff, to provide enough time to complete all the required steps and procedures. This time, of course, is not realistic for all steps - only for the last seconds. With realistic times, the users would have to sit in front of the monitor for hours, waiting for something interesting to happen!

Automated sequence may be switched off (and later back on) by the **M** key (stands for "go **M**anual") at any time until the last few seconds. If you want to start the scenario with the automated launch sequence disabled, specify the **MANUAL_CONTROL** variable in the launchpad vessel section of the scenario.

Also in the scenario, you specify the exact takeoff MJD moment with the **LAUNCH_MJD** scenario variable. For better "playability", I recommend setting the starting MJD time of the scenario not later that the 100 seconds required to complete the launch sequence - but not too far in advance as well. The best time will be about two minutes prior to takeoff.

Automated launch sequence performs a set of sanity checks when scenario is loaded - and will not initialize, if any of the checks fail. The checks take into account the time constraints, the presence of the flight program, the suitability of the specified flight program for the payload installed on the rocket, and so on. If the sequence has failed to initialize, you'll get a screen message explaining the scenario error.

Regardless of the launch mode, if you have focused the launchpad "vessel" at the time of the launch sequence, the focus will automatically switch to the launch vehicle 15 seconds after the launch.

Bypassing the automated launch sequence

Two minutes for the launch sequence is not much, but, if you are doing a lot of flight tests, you may want to bypass it completely and launch the rocket "right away". You can use **L** key for that (stands for "**L**aunch now"). This key "warps" the launchpad to the launch configuration (if this is not done, the rocket may hit something and explode on takeoff), and the engines are ignited.

One thing to remember here. if your launch timing for the mission is crucial for space interception or rendezvous, you'll likely miss it because you take off earlier than planned. In this case, it is better to speed up the launch by the usual Orbiter "time warp" (the launch works well in time speeds up to x10).

Another thing to remember: The bypass feature is completely independent of the auto/manual mode. It will work in both. But if you launch in manual mode, the rocket will inherit the "off" state of the autopilot, and... well, the rest depends on you.

Targeting the rocket

The easiest way to set the particular ballistic or orbital trajectory for R-7 is to use the scenario variables.

For ballistic trajectory, you can use **TARGET_NAME** variable to set one of six pre-defined targets: **Kura** (a Russian test range), **NewYork**, **Washington**, **Chicago** and **LosAngeles** (Caribbean crisis targets) and **Canaveral** (a good built-over demo target).

To point the rocket to a custom target, specify the target coordinates as follows: **TARGET_POS XX.XXX YYY.YYY** where x stands for latitude and y stands for longitude, in degrees, in +- range. But remember, that the accuracy for the custom target may not be as good as for the predefined targets. This is a historic reconstruction addon, not a space shooter, so I did not try to create universal ultra-accurate ICBM guidance system!

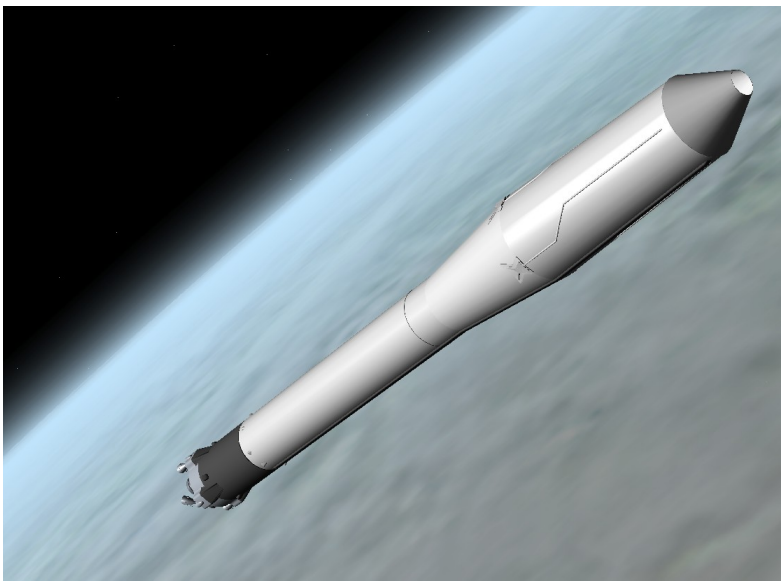
For orbital trajectory, specify the target orbit as follows: **ORBIT PPP.PPP AAA.AAA II.III** where P is perigee in km, A is apogee in km, I is inclination in degrees, in +- range. Once again, the final orbit may end up to be slightly off, which was not unusual in these early rocketry days.

Some basic sanity checks are implemented for the flight program setup. If the trajectory parameters are invalid, unreachable or unsuitable for the installed payload, you will get an error message when loading the scenario, and the automated launch sequence will not initialize.

Alternatively, you can specify the new flight program or change the existing program (along with the remaining takeoff time) interactively. To do so, switch to the "cockpit/bunker" view of the launchpad "vessel" and open the targeting panel by the **F8** key.

Use mouse clicks on green triangles to set the numbers. Once satisfied, click the "**Set Flight Program**" button to load the new flight program into the rocket and initialize the launch sequence.

Block A (Core Stage, Second Stage)



Block A is the main rocket module, the core stage of R-7. In Russian convention, the modules were named by the letters of Russian alphabet - where 'A' is also the first letter. At the same time, in the Russian convention of *numbering* the rocket stages, block A is considered the second stage: it continues to fly when the first stage (the four strap-on side blocks) are already discarded.

In fact, in this original two-stage version of R-7, the core stage flies all the way through the mission, from start to end. And then it travels together with the warhead to its destination, or it orbits the Earth with the satellite it just launched.

When the core stage completes its mission and separates the payload, it performs the final step of the flight program: opens the valve to dump the pressure from its fuel tank through a special nozzle on its side. This vent sends the rocket tumbling, which helps it to break up into smaller, less hazardous pieces upon reentry. I don't know if this practice still went on with the tree-stage R-7-based rockets (including today's Soyuz).

In one special case, though, the last phases of the flight were different. The second satellite, that was carrying a dog passenger, was designed to stay attached to the core stage, in hope to improve the heat dissipation. For the same reason, the final pressure dump in this flight was symmetrical, to prevent core stage tumbling. As yet another difference, MECO in this launch was conducted after complete fuel consumption, rather than at any particular pre-programmed conditions.

In two-stage version, the core stage carries a payload adapter, specifically designed for a particular type of payload. In this addon, the rocket auto-detests the payload and loads the appropriate adapter mesh. If you want to launch your custom payload on R-7, and the default adapter does not fit, you can specify your own custom payload adapter mesh in the **ADAPTER_MESHNAME** scenario variable.

In the "cockpit view", the core stage is equipped with two rocketcams. The default rocketcam looks down, the other looks up. To switch the view between them, use **C** key.

As the core stage is active through the full powered flight, it is naturally responsible for piloting the whole assembly, either in autopilot (default) or in manual control mode. The autopilot receives the flight program from the launchpad "vessel". It can reliably fly with the time acceleration of up to x10. Time acceleration will be reset to x1 as the rocket approaches the final phase of the flight, shortly before MECO and payload separation.

To switch between autopilot and manual control, use **M** key. If you are taking off from the launchpad in the manual mode, simply power up the rocket. As soon as the thrust exceeds the weight, the rocket is released, and the launchpad supports and cable masts fall away. But be sure to lower the service towers prior to manual takeoff - they do NOT fall away automatically, and any attempt to fly through them will be disastrous!

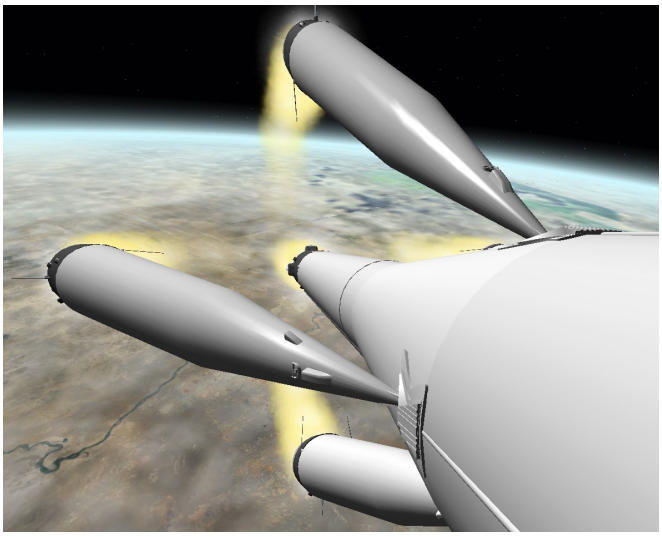
When in manual mode, you can jettison the attached elements with the **J** key. The elements are separated with the repeated key press in the following order: strap-ons (if still attached), fairing (if any), payload (warhead or satellite, except for the historical second satellite).

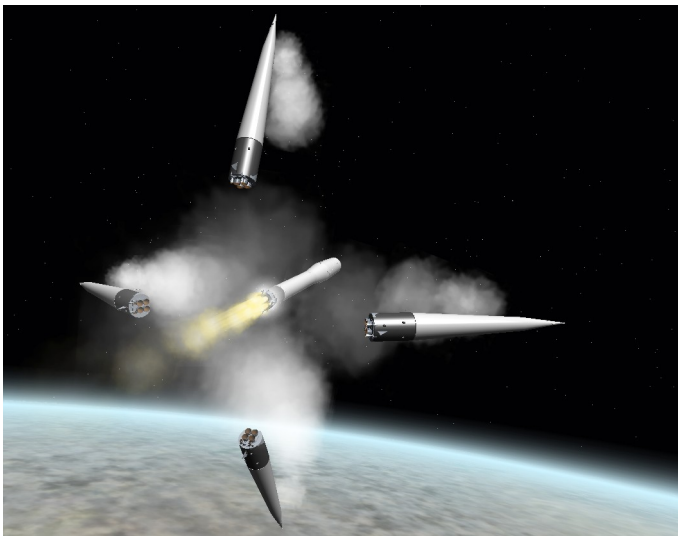
When payload is separated, whether automatically or manually, the focus by default gets transferred to the payload - because in most cases this is exactly what you want. However, sometimes you may want to "stay" with the launch vehicle through the separation process to get interesting separation visuals. To disable automatic focus switching, use **Ctrl+J** key.

For the description of the attachments parameters of payload and fairing, see the 'Launching your own payload on R-7' section.

Final note is about the rocket reliability. By default, the core stage reliability is limited, and from time to time it can encounter all kinds of random failures, including some disastrous events. If you plan to use the rocket in the custom scenario with the controlled reliability, set the **RELIABILITY** variable in the scenario for the core stage (and for all strap-ons) to the required percentage (a floating point number between **0.** and **100.**)

Blocks B-D (Strap-ons, First Stage)





R-7 used four strap-on blocks, which, in Russian staging classifications, are considered the full first stage, not the strap-ons. Following the letters of Russian alphabet convention, they are called Block B, Block V, Block G and Block D. This naming convention was used on many Russian rockets, and sometimes leads to confusion.

Putting the first stage to the sides of the second was beneficial for many reasons in those early days. First, it reduced the bulk of the individual rocket modules, to make it easier for transportation. Second, it provided the ground ignition of both stages, eliminating the need or in-flight ignition - a huge boost to the overall reliability in times when even the ground ignition was far from perfect!

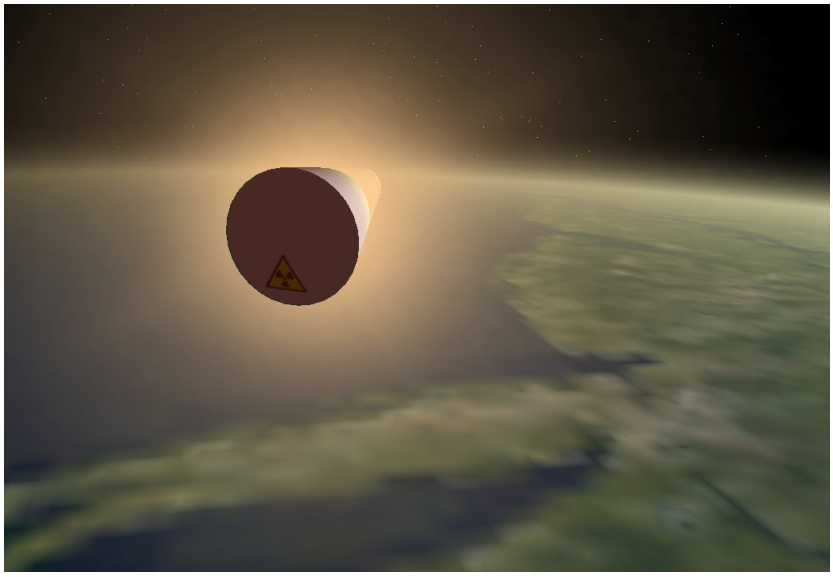
The ingenious "separation system" of the side blocks did not involve any special separation motors. It relied exclusively on the regular flight hardware: normal rocket thrust to rotate around the top mount point, and fuel pressurization vent to bring the top end of the block away from the core body. Not much to fail!

Speaking of failures. Structurally, the code body "stands" on the side blocks. Thus, is the thrust of any of the side blocks drops significantly due to some engine problem, this side block cannot support itself, and simply drops out of the packet. This quickly destabilizes the remaining strap-ons, and eventually the whole packet falls apart. This unfortunate but spectacular scenario was a signature failure of the early R-7s for quite a while...

In terms of control and configuration, not much is to be said about the strap-ons. In this addon, just like in real life, they are fully controlled from the core stage. The only individually configurable parameter is the **RELIABILITY** scenario variable that allows to set the likelihood of failure separately for each block. Correspondingly, if you want to avoid being knocked out by the failure, you have to specify "**RELIABILITY 100.**" in the scenario for each of the four strap-ons - as the default is less than 100%.

By default, the focus on side blocks is disabled - as wished by most beta testers, to keep the list of the vessels shown by **F3** key as short as possible. You can enable focus on them (and on other unfocusable vessels in the scenario) by pressing **Ctrl-F** in any other vessel. You may want to switch to any of the side blocks to get some interesting visuals, especially in the active phase of the flight. For the same reason, a rocketcam view is implemented as the "cockpit" view of side block.

Warhead



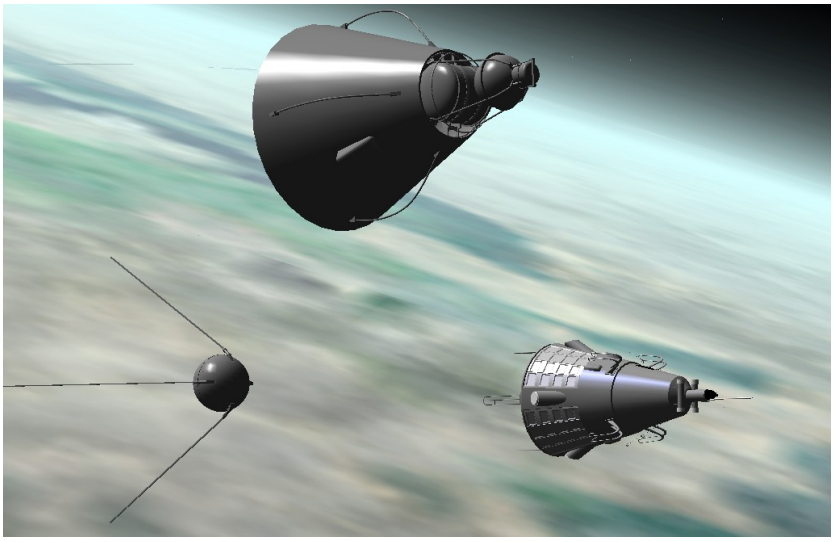
It is important to remember that all great technical achievements of the 20-s century started as deadly weapons. Spaceflight was no exception. Before (and after) opening the Space Era for all mankind, R-7s was carrying military warheads. Luckily, only test items - but at least once they came to the very edge of delivering The Real Things...

This addon does not skip this "darker" side of the rocket's history. You can read more about the military history and the different types of R-7 warheads in the 'R-7, the first ICBM' section of this manual. You don't have to worry about the selection of the warhead type, though - the proper configuration is loaded automatically, based on the year of the simulated scenario.

Warhead is a completely passive vessel. The only control parameter you can apply to the it is the **NUKE** scenario variable that equips the warhead with a 3-megaton nuclear device. This nuclear device gets armed far into the flight, so a launch failure would not lead to the "accident" over your own territory.

Warhead hosts a number of additional visual effects for glowing red in the upper atmosphere on reentry. One of these effects causes Sun to temporarily disappear from the scenario (due to the documented bug in Orbiter 2006 P1). This bug is reported to be fixed in new beta versions of Orbiter.

First Three Satellites



The first three satellites launched by R-7 are so frequently referred to together, as a series, and as such are groped together here as well.

They were not serial satellites, of course. Each one was unique, and had it own unique design history.

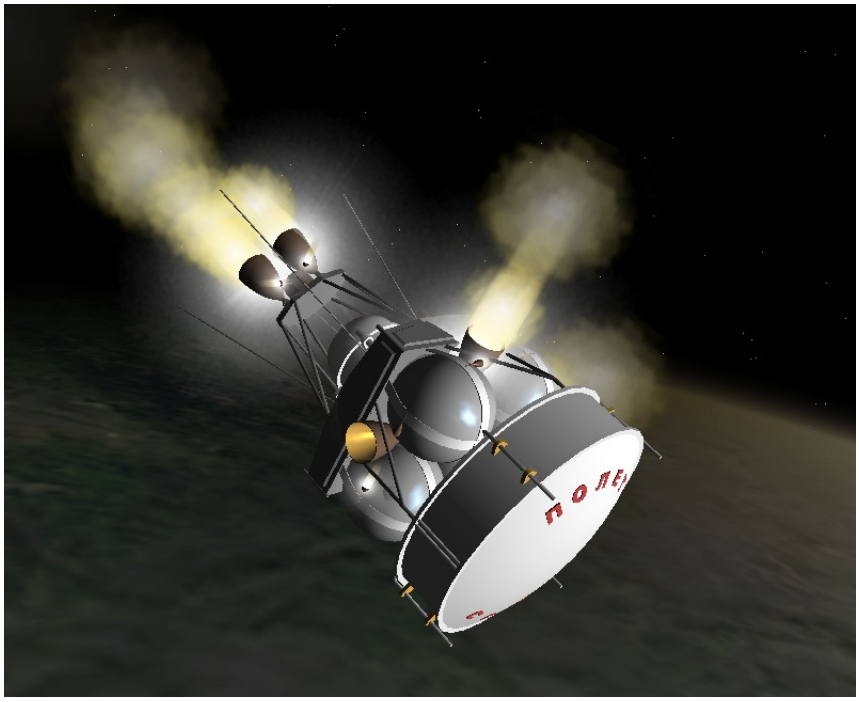
The first satellite was a simple radio beacon, simply to verify that it indeed went around the globe. It was transmitting the signal as long as the batteries lasted. It was also reported to be seen by many observers all over the world, though in most cases what people saw was not the satellite itself, but the huge core stage travelling in the same orbit.

The second satellite, built in record time after the first, was carrying a passenger, a small female dog named Laika. This was a one-way trip, that had risen some controversy among the dog-lovers, though even nowadays the impatience of the space industry leaders of those early years can be easily understood. To give them a credit, they had thought as well as they could to maximize the gains from this honorable sacrifice, and to avoid any unneeded suffering. From the design point of view, it is interesting to note that the life support system for Laika had not to be designed from scratch - it was already standard, taken as-is from the previous high-altitude rocket flights of other dogs. Another notable feature of the second satellite was lack of the separation system. The satellite remained attached to the rocket, in hope to provide better dissipation of excessive heat. For the same reason, the vent system of the rocket itself was modified for symmetrical pressure dump, to avoid tumbling.

The third satellite was carrying many scientific instruments for space research. It was also more heavy - and more sophisticated: equipped with active thermal control radiators and even solar panels. It was this complexity that delayed its development: originally, it was planned to be the first satellite, not the third.

In this addon, the satellites can't do much rather than be deployed into the orbit. They simply carry radios that play authentic transmissions (you can toggle radio playback on and off with the **B** key). Of all three satellites, only the second implements a cockpit view that is worth looking at (well, the other two did not have cockpits).

Polyot



Polyot was in many ways unique spacecraft! Very secretive, too. Up to this day, only a few accounts and murky photos are available, with no detailed information. So recreating this model for the Orbiter had involved some guesswork.

It is known that the spacecraft was carrying a lot of fuel for its test maneuvers, and a lot of engines facing in all directions. The final, operational version (launched on different kind of rocket) was carrying an explosive charge to destroy its target (after all it was an ASAT system!) It is not known if Polyot was equipped with one, so I took the freedom to do so.

Polyot was not tested against a real target. But we can do it in Orbiter, just to make the mission more fun and have a goal for the flight. Of the two supplied scenarios, Polyot-2 mission contains a target (a spend R-7 core stage in LEO), and has a goal of hitting it with Polyot. If you want to set the target hit as mission goal in some custom scenario, specify the **MISSION_HIT** scenario variable for Polyot in the scenario file.

For the purpose of ASAT function, Polyot spacecraft in this addon implements a very simple autopilot. Here are the control keys for the autopilot:

Space - select the next target in the list of possible targets;
L - lock on the selected target;
K - kill the selected and locked target.

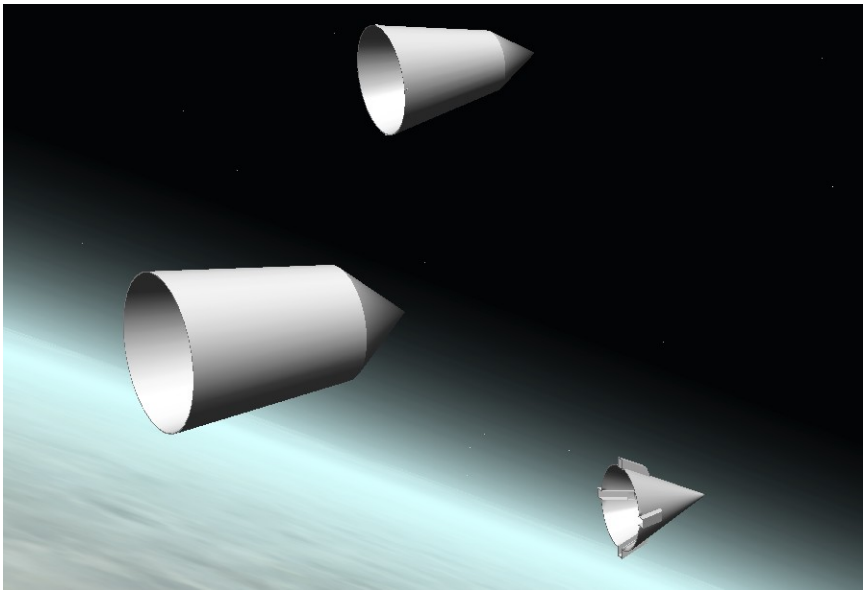
To be selected as a target, the vessel should be flying (not sitting on the ground), should allow to be focused, and should be within 10 Mm from Polyot.

To be locked on, the selected target vessel should be within 50 km from Polyot.

You can re-target Polyot at any time, or cancel any autopilot mode by using the same key as toggle. In case of the miss, the autopilot mode is reset.

In case of successful hit, it may be interesting to make one full orbit around the Earth with any of the wrecks, and then look at the resulting orbital distribution of the new "space junk" in Google Earth.

Fairing



Payload fairings used on the original R-7 were different from what we usually see today. They were not discarded halfway to the orbit, like the modern fairings, but stayed with the rocket all the way up. They were jettisoned only shortly before the payload itself, in zero-g. Because of that, they did not have to be two-part shells, like the ones we see today, but rather were simple single-shell designs.

The fairings were custom-made for each launch in those early days, to fit each custom payload and payload adapter. In this addon, the rocket figures out which fairing configuration to use, depending on a particular payload attached. If you want to launch your own payload on R-7, its type won' t be recognized by rocket, and a default configuration will be loaded. If it does not fit your payload, specify your own fairing mesh with the **MESHNAME** scenario variable.

To properly position your fairing on the rocket, see instructions in the " Launching your own payload on R-7" section of this manual.

By default, the focus on fairing is disabled - as wished by most beta testers, to keep the list of the vessels shown by **F3** key as short as possible. To get some interesting custom visuals, you can enable focus on it (and on other unfocusable vessels in the scenario) by pressing **Ctrl-F** in any other vessel.

Other Vessels in Scenarios

You may encounter some other vessels in the supplied scenarios or in the saved scenarios:

kmlWriter

This "hidden" vessel does not have any visual representation. It quietly sits in the scenario, collects information about positions and important events from all other vessels, and writes it into Google Earth file.

Technically, it could have been implemented as Orbiter Launchpad plugin, but there are a few benefits in not doing it. First, being a "vessel", it can use my new cross-vessel communication infrastructure. Second, it does not require any user actions to be activated. Third, it does not create the illusion that it can work with "other" vessels that do not belong to this addon (it cannot).

Wreck

Almost all vessels in this addon can break apart. When they do so, they produce wrecks. A wreck is a wreck, it can't do much. Some smoke and flame visual effects (when in atmosphere), some barely recognizable metal pieces - that's about it.

Wrecks are fully traceable both in Orbiter and in Google Earth. As such, they become more useful than one may initially think. They can be used to visualize the spread of debris fields after launch failures, to evaluate dangerous fallout areas, to verify impact sites for the jettisoned hardware, to see the "space junk" orbits after successful ASAT tests, etc...

Wrecks inherit the "focus" behavior from their originators. So, even if the focus of the wreck is disabled by default, you can always toggle it with the usual **Ctrl+F** keys through the scenario.

Depending on the situation, a vessel can produce one or more wrecks. Wreck names always keep reference to the original vessel, so you can always tell which wreck you are looking at.

Launching your own payload on R-7

This addon is made as a historic reconstruction rather than a universal general purpose launcher. However, it is possible to use this implementation of R-7 to launch any custom payloads within its physical capabilities.

Custom payloads are launched into orbital trajectory, not to ballistic trajectory. You'll have to adjust payload and fairing attachment parameters on Block A vessel in scenario (see supplied scenarios for samples). For payload attachment, use the following scenario variables: **PAYLOAD**, **PREF**, **PDIR**, **PROT**. For fairing attachment, use the following scenario variables: **FAIRING1**, **FREF1**, **FDIR1**, **FROT1**. In these sets, the first variable specifies the vessel name of the attached object, the other three variables are the 3D vectors for the attachment point coordinate, direction and rotation *on the object being attached*. The default setting for these vectors are (0,0,0), (0,0,-1) and (1,0,0). If any of your settings are the same as the default ones, you don't have to specify them.

You also may have to produce your own meshes for payload adapter (**ADAPTER_MESHNAME** in Block A vessel) and fairing (**MESHNAME** in Fairing vessel).

Custom payloads will not benefit from some of the advanced features implemented specifically for the vessels included in this addon. For example, they will not be plotted in Google Earth, they will be bypassed in the **TAB** or **Shift-TAB** focus switch order, they won't disintegrate in realistic manner with the rest of the vehicle, etc.

Launching early R-7 payloads on other rockets/addons

Can you launch any of the payloads, provided in this addon, on other rockets? I don't know. This has never been tested, and had never been designed for.

It may turn out to be more problematic than launching "other" custom payloads on "this" R-7. All payloads in this addon are designed to interact with the launch vehicle, and if the launch vehicle is not responding as expected, the results may be... well, unpredictable. But, if never try, you never know...