

## 1, 2, 3, кодирај ! – 5-6 разред основне школе - Етапа 2.6: Избегавање препрека, вођење рачуна о броју живота

Резиме	Ученици комплетирају свој програм уводећи препреке које је потребно избећи (нови ликови) и креирајући променљиву за свој број « живота ». Преиспитују појмове теста, петље и претходно поменуте варијабле, и продубљују појам догађаја.
Појмови	<p>« Машине » :</p> <ul style="list-style-type: none"><li>• Машине које се налазе у нашем окружењу само извршавају наредбе (инструкције).</li><li>• Комбинацијом више елементарних инструкција омогућујемо им да изврше комплексне задатке.</li></ul> <p>« Алгоритми »</p> <ul style="list-style-type: none"><li>• Алгоритам омогућује да решимо неки проблем.</li><li>• Петља омогућује да се нека иста активност понови више пута</li><li>• Неке петље се никад не заустављају па носе назив « бесконачне ».</li><li>• Неке петље се понављају одређен број пута и називају се « итеративне ».</li></ul> <p>« Језик » :</p> <ul style="list-style-type: none"><li>• За давање инструкција некој машини користимо неки програмски језик, разумљив и машини и човеку.</li><li>• <i>Скрач</i> радно окружење графичког програмирања користи једноставан језик.</li><li>• Неки програм је представљен алгоритмом у неком програмском језику.</li><li>• Неке инструкције се извршавају само кад се деси неки догађај па говоримо о програмирању вођеног догађајима.</li><li>• Неке инструкције се извршавају једна за другом па кажемо да је реч о секвенцијалном програмирању.</li><li>• Извршење неког програма је репродуктивно (ако се не мењају ни инструкције ни податци којим манипулишемо, онда програм даје увек исти резултат)</li></ul>
Материјал	За сваки пар

- |  |  |
|--|--|
|  | <ul style="list-style-type: none"><li>• Компјутер са <i>Скрачом</i> и програм меморисан на претходном часу</li><li>• Нови ликови кје ћемо користити на овом часу, преузмите их <a href="#">овде</a>.</li></ul> |
|--|--|

**Педагошка напомена:**

Можда би могло бити од користи да, с времена на време, прикажете « финалну » верзију игре (верзија коју је правио наставник), да би повећали мотивацију ученика и указали на оно битно што следи у следећој етапи. Овде се ради о једноставној демонстрацији игре, а не о приказу решења и читању програма.

Наставник, у настојању да што више зачини игру којом симулира мисију у свемиру, уводи препреке (језеро лаве, пешчану дину) и нову променљиву која се односи на број « живота ». Ровер покреће игру с 3 живота, враћајући се у базу, сваког пута када додирне неку препреку губи један живот. Игра се зауставља када је број живота 0.



### Задатак 1 : увођење нових ликова (5 минута)

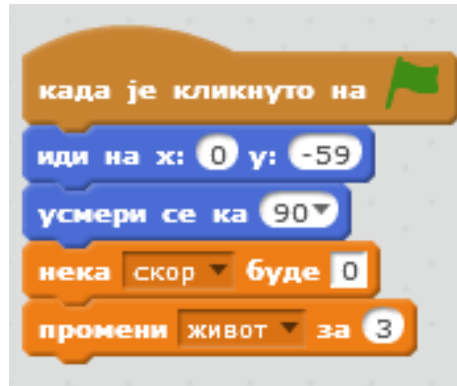
Ученици, користећи оно што су научили током претходних часова, уводе без икаквог проблема 3 нова лика (база, пешчана дина и лава), постављају их тако што фиксирају њихове позиције у неком делу екрана. На пример, могу да одлуче да базу поставе у центар а роверу ће дати другу почетну позицију (јер је он претходно био у центру).

Можемо, али то није неопходно, да јасно прецизирамо све о различитим ликовима (ровер, база, дина, вегетација, лед). Одлука о томе који од ликова ће бити у првом плану реализује се инструкцијом « **пређи сасвим напред** » која се налази у категорији « **изглед** »



### Задатак 2 : дефинисање и давање вредности променљивој « број живота » (5 минута)

Веома лако креирамо нову променљиву « животи » којој дајемо вредност 3 (на истом месту као што смо радили при дефинисању скорa, на пример у програму ровера). Програм за ровера од сада садржи (и подпрограма који омогућује да га водимо) :

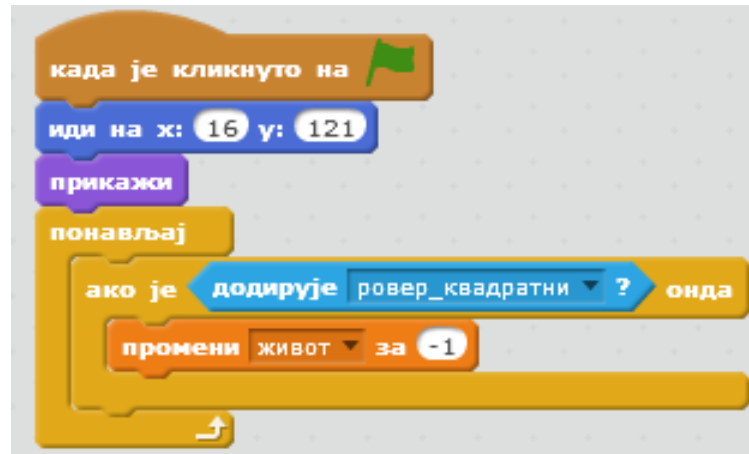


### Задатак 3 : када ровер додирне лаву губи један живот (30 минута)

На претходном часу се променљива « скор » повећавала за 1 када је додирнут неки ресурс. Овде се при додиру лаве или дине променљива « живот » смањује за 1. Саветујемо вам да прво решите проблем са једном препреком (лава), а затим то поново урадите али за другу препреку (дина).

#### Како одузимати у *Скрачу*?

Ученици покушавају да нађу како да контролишу/програмирају одузимање. Постоји само команда « додај » али не и команда « смањи ». Дискутујте заједно, ако закључите да је потребно, да се овај проблем може решити тако што ћете за одузимање 1 додати вредност -1.  
Програм за лаву је:



Провизорни програм за лаву

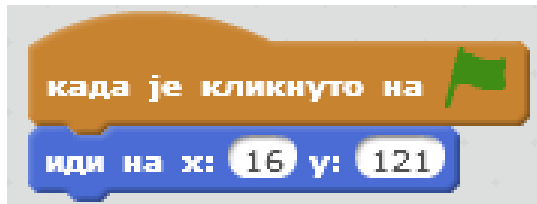
Међутим, овај програм има недостатк, јер када ровер додирне препреку они остају извесно време у контакту (неколико десетинки секунде, или неколико секунди ако се више не креће) и у току тог времена променљива « живот » се смањуј. После само неколико секунди негативна вредност живота је око -4000.... Једини начин да се то заустави је тренутни прекид контакта између ровера и препреке.

Пошто је фиксиран препрека, потешно је померити ровер. Ово се не може урадити у програму за лаву или дину, јер мора се бити у програму за ровер да би му се давале команде за позицију.

## Како померати ровер?

Постоје бар два могућа решења овог проблема:

- Решење 1  
Елиминишите програм за лаву (изузев почетне позиције) и онда направите програм сличан програму за ровер. У програму за ровер додајемо, на пример, команду врати се у базу.  
Имајмо на уму, да за повратак ровера у базу, можемо дати наредбу иди у (X=0, Y=0), или да му кажемо иди у позицију lika « база ». Последње решење је боље јер ће он ићи у њу чак и када је база постављена на неком другом месту.



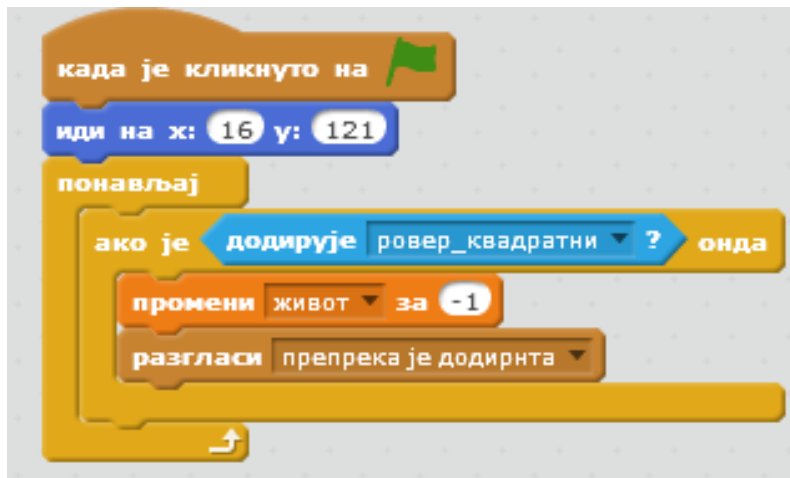
Програм за лаву



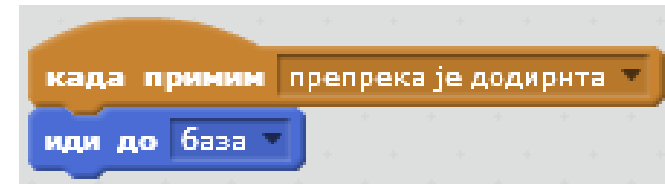
Програм за ровер

Решење 1

- Решење 2 (елегантније... и практичније за наставак)  
Знатно боље решење се састоји у очувању програма који смо направили за лаву, уз додавање нове инструкције која ће послати поруку другим програмима (посебно програму за ровер). Ова порука ће у програму за ровер покренути активност (врати се у базу). Исто као код имена променљивих, назив поруке мора бити експлицитан. На пример, наша порука је овде « препрека је додирнута ».



Програм за лаву



Програм за ровер

## Решење 2

Програм лаве шаље поруку другим програмима ( роверу, пре свега).

Пријем ове поруке у програму за ровер је догађај који покреће активност (иди у базу) Команде, које омогућују да шаљемо поруку или да покренемо неку активност када је та порука примљена, се налазе у категорији « **догађаји** ».

### Педагошке напомене:

- Можемо да замислимо и другачије решење које би се састојало у програмирању ровер тако да се он одбије када дотакне неку препреку (у овом случају контакт не траје).
- Друго решење, које се може сматрати елегантнијим, се састоји у инсистирању на какртеру програма вођеног догађајима. Програм може да шаље поруку, коју ће користити други програми, сваки пут кад се појави неки догађаја (на пример, додирне препрека). Појам догађаја је већ уведен током претходних часова (« када је горња стрелица додирнута », « када је зелена застава додирнута »...), али у овом случају користимо по први пут нови тип догађаја који се односи на пријем неке поруке коју шаље неки програм.
- Препуштамо наставнику да користи једну или другу методу у зависности од тога да ли жели да свој час посвети ревизији већ уведених појмова или пак жели да уведе нови појам који се овде односи на поруку. Сасвим је сигурно да мора да води рачуна о идејама ученика и колико и како су савалдали претходне појмове!
- Напомена: слање и пријем порука ће бити поново коришћено при завршетку игре (види « **game over** »).



## Задатак 4 : поновити све али сад за случај пешчане дине (10 минута)

Пошто су ученици успели да воде програм са једном препреком (лава), требало би сад да ураде исту ствар али са (пешчаном дином). Ова мала вежба би омогућила да утврде оно што су претходно урадили, пре свега слање и пријем поруке.

### Закључак у форми тескта

Ученици допуњавају листу инструкција које су које су користили и знају у *Скрачу*.

---

<< [Etape III-2.5](#)

[Séquence III-2](#)

[Etape III-2.7](#) >>

Extrait de "[1, 2, 3... codez !](#)", Editions Le Pommier, 2016-2017. Publié sous licence [CC by-nc-nd 3.0](#).