

1, 2, 3, кодирај ! –Активности циклуса 4 – Пројект « Аркадне игре » - Час 4: Програмирање падања клопки





Доминантна дисциплина	Математика
Резиме	Ученици употребљавају варијабле за програмирање убацивања клопки (брзина падања клопки, и броја преосталих живота). Поново користе условне структуре АКО...ОНДА.
Појмови	<p>« Машине » :</p> <ul style="list-style-type: none">• Варијабла је име које дајемо некој зони меморије. Она омогућује стокирање нечег, а затим и његову каснију поновну употребу, или модификацију. <p>« Језик » :</p> <ul style="list-style-type: none">• Неке инструкције се извршавају симултано у односу на друге, реч је о програмима који се реализују истовремено. <p>« Добра пракса програмирања » :</p> <ul style="list-style-type: none">• У програму је пожељно да користите веријабле а не да стављате нумеричке вредности.• Варијаблама морате дати име.• Пожељније је још на самом почетку дефинисати почетну вредност варијабле.• Увођење пауза током извршења програма олакшава проналажење и елиминисање багова.
Материјал	<p>За сваки пар ученика</p> <ul style="list-style-type: none">• Компјутер с доступном интернет везом (за коришћење on-line верзије <i>Скрача</i>) или са претходно инсталираним <i>Скрачом</i>• Свеска за пројект <p>За наставника</p>

	<ul style="list-style-type: none"> • Компјутер с доступном интернет везом (за коришћење on-line верзије <i>Скрача</i>) или са претходно инсталираним <i>Скрачом</i> • Видеопроектор
--	---

Педагошке напомене:

- На овом часу се ученицима омогућује да се сроде с врло важним појмом варијабли (с којим се срећемо у свим програмским језицима).
- У припреми часа наставник мора да изврши поделу на једноставне задатке функционалности које се програмирају, и предложи мапу пута (према доњој табели). Ученици ће током часова који следе стећи више искуства па ће моћи и сами да разложе пројект на једноставне задатке, у почетку групно а затим и сваки поједини пар ученика.
- Илузорно је, од овог тренутка, тражити да сви ученици напредују истим ритмом (јер ће се групе које напредују лакше врло брзо досађивати и раставити). Наставник лакше води одељење захваљујући подели на етапе и задатке јер сваки има нешто да ради.

Одељење преузима менталну мапу која је направљена током 2 часа и интересују се за основне механизме игре: падање астероида (или других типова клопки зависно од теме коју су изабрали ученици). Саветујемо вам да усмераваате ученике на следећи начин (идеално би било да истовремено дискутујете с ученицима) :

Тежина	Назив задатака	Рад који је потребно реализовати
	Задатак 1: прикажите падање 1 астероида с неке дефинисане позиције	<ul style="list-style-type: none"> • Креирање лика • Његово позиционирање у горњем делу екрана • Програмирање пада ка доњем делу екрана
	Задатак 2: прикажите падање 1 астероида с неке случајне позиције	<ul style="list-style-type: none"> • Урадите тако да астероид никад не полази са истог места када покренете програм.
	Задатак 3: прикажите губитак живота аватара када додирне неки астероид	<ul style="list-style-type: none"> • Креирати варијаблу која се односи на број преосталих живота • Дајте почетну вредност • Смањујте је за 1 кад год аватар буде погођен астероидом
	Задатак 4: користите клонове за приказивање падања више астероида	<ul style="list-style-type: none"> • Креирати више клонова астероида • Направите да ови клонови падају у случајном временском интервалу

		<ul style="list-style-type: none"> • Направите да број клонова буде случајан број • Направите да клонови нестају кад дођу у доњи део екрана
--	--	---

NB : задатак 3, као и почетак задатка 4 се реализују заједнички. Задатак 4 је описан на следећем часу (непотрбно је уводити толико нових ствари на једном часу!)

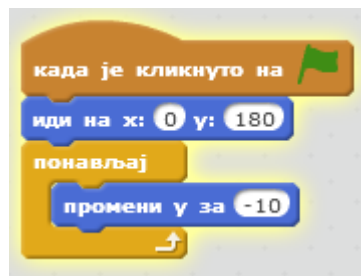


Задатак 1: креирање падања 1 астероида са дефинисане позиције (10 минута)

Ова етапа неће ученицима представљати неки проблем јер су већ користили све функционалности које омогућују програмирање.

- Потребно је на почетку креирати нови лик за астероид (ово се може урадити цртањем тог лика, или његовим преузимањем из неке од библиотека *Скрача*, или из неког фолдера).
- Позиционирање лика у горњем делу екрана се реализује помоћу инструкције « иди на x... y... » (у категорији « Кретање », плава боја)
- Падање астероида према доњем делу екрана се реализује модификацијом ординате.

Могући програм је дакле:



NB : овај програм је могуће модификовати на више начина које у наставку приказујемо

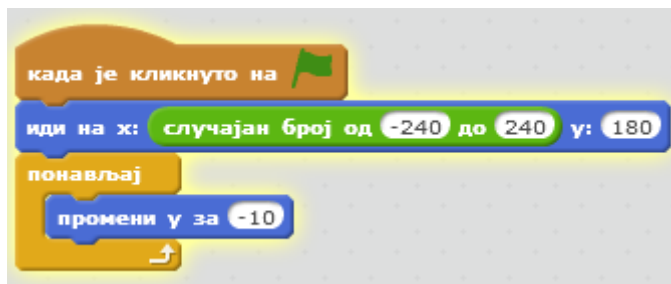
Наставник организује кратко заједничко представљање резултата да би се уверио да је овај задатак реализован јер ученици немају још увек довољно искуства. Посебно наглашава да, у *Скрачу*, сваки лик има свој програм. Различити програми различитих ликова се реализују паралелно пошто је сваки од њих покренут неким догађајем (за сад, једини догађај који смо упознали је клик на зелену заставицу ... али ћемо ускоро видети и друге догађаје!).



Задатак 2 : креирање падања 1 астероида са случајне позиције (15 минута)

Наставник подсећа, враћајући се на претходно формирану менталну карту, да астероид непрекидно пада с неке неодређене позиције (која је увек у горњем делу екрана). Од ученика тражи да упознају различите делове у простору за скрипте јер се ту налазе инструкције које могу да користе при креирању случајног броја, затим и за одговарајућу модификацију програма.

Решење се налази у делу « операције », па лако модификовани програм изгледа овако:

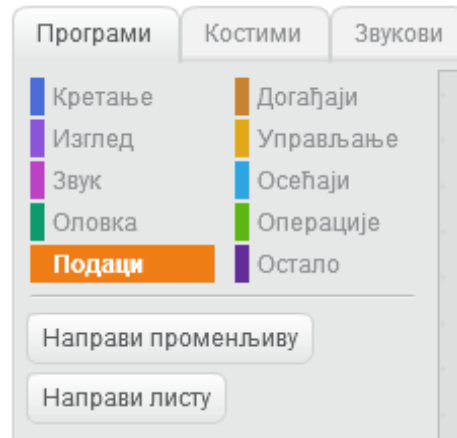


Задатак 2 : креирање губитка живота аватара када додирне астероид (20 минута)

Пожељно је да ову етапу урадите заједнички јер ученици нису до сада радили с варијаблама.

Наставник подсећа ученике да аватар поседује само одређени број « живота » (на пример, 3 на почетку игре). Евиденција о броју живота се води посредством неке « варијабле »: варијабла је уствари неки део меморије компјутера који може да садржи неку вредност (ова вредност може да се мења).

Показује ученицима како се креира нека варијабла (клик на « направи порменљиву » у делу « Подаци », оранж боје).

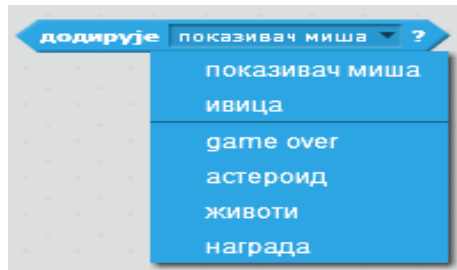


Научне и педагошке напомене:

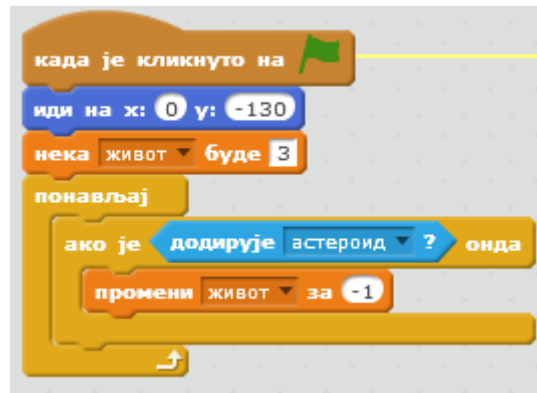
- Тако креирана варијабла може бити доступна само једном једином лику (оном који је у програму креиран) или свим ликовима. Овде је реч о локалној и глобалној променљивој у другим језицима програмирања. Пошто још не знамо да ли ће други ликови имати потребу да користе ову променљиву онда је мудро да је учинимо доступном свим ликовима.
- Да би програм био разумљивији, неопходно је да дате експлицитно име варијаблама које сте креирали. Овом добром праксом смањујете могућност багова у програму. Дакле име ове варијабле може једноставно да буде: « животи ».
- У нашем програму ће бити више варијабли (број живота, скор, брзина падања клопки и награде ...). Зависно од оног што вам више одговара, логично би било да групишете ове варијабле у исти програм, који сматрате главним програмом (на пример, програм за аватара), или пак да их креирате и уведете у програме који су у вези с њима. Овде се, варијабла « животи » односи на лик « летилица »...па изгледа логично да је креирамо и уведемо у програм лика « летилица ».
- Напомињемо да се, када је варијабла креирана, на екрану појављује њен назив као и њена вредност. За њено елиминисање с екрана довољно је кликнути на квадртић на левој страни имена варијабле у делу « Подаци ».
- Наставник напомиње ученицима да су већ манипулисали променљивим на претходним часовима (апсциса X и одрината Y, које су одређивале позицију одабраног лика на екрану). Ове променљиве су већ биле доступне ученицима па су зато имали могућност да с њима манипулишу (праве тестове, мењају њихове вредности ...) али нису имали потребу да их креирају.

Ученици креирају ову варијблу, дају јој име « животи » и почетну вредност (на пример 3).

Поново користе условну (или контролну) инструкцију АКО... ОНДА с којом су се већ срели на часу 3 (она се иначе налази у делу « Управљање » жута боја), као и део « Осећаји » (светло плава боја) који дају назнаку да ли се ликови « астероид » и « летилица » сударају. Ако ову функционалност додамо у програм за летилицу онда ће инструкција из дела осећај имати следећи облик:



Главни програм за летилицу тад постаје:



Провизорно решење за вођење рачуна о броју живота.

Педагошке напомене:

- Постаје нам јасније да је потребно тачно именовати не само варијабле него и ликове. Када радимо са више ликова онад је то обавезно. Инструкција « ако је астероид додирнут » је знатно јаснија него « ако је лик2 додирнут »!
- За преименовање ликова потребно је да прво одаберете, а затим и кликнете на « i » која се појављује у плавој боји на икони лика.
- Овде смо одлучили да у програму за летилицу узмемо у обзир и њен судар с астероидом. То изгледа знатно логичније у овом тренутку (јер је у питању вођење рачуна о броју живота летилице). Касније ћемо видети (Задатак 2 [Час 7](#)) да, ако желимо да побољшамо графички изглед игре (промена изгледа летилице када при судару нестане астероид), једино решење је да овај тест поставимо у програм за астероид. Професорима саветујемо да, већ сад, не намећу ученицима оно што би требало да буде *in fine* добро решење, него да следи логику ученика а затим и одустану од модификације програма.

Међутим, врло брзо се суочавамо са следећим проблемом, ликови « астероид » и « летилица » се сударају, а тај судар траје извесно време, тј., не дешав се тренутно. Током тог временског интервала се варијабла « живот » стално смањује. При сваком додиру се број живота смањује, што пак не одговара оном што смо желели.

За идентификацију овог проблема можемо увести, на пример, паузу од 1 секунда унутар петље « понављај ». На тај начин ћемо лакше пратити извршење програма « корак по корак » и јасно видети како се вредност варијабле мења. Уопштеније речено, **увођење пауза у неком програму је добар начин да се одстрани багови у програму.**

За решење овог проблема довољно је да се контакт 2 лика прекине. На пример, реализујемо нестајање летилице током $\frac{1}{2}$ секунде коришћењем инструкције « сакриј » (део « изглед », љубичаста боја). Пажња, пошто захтевамо да се она у једном тренутку сакрије, потребно је преузети обавезу да се реализује њена појава при покретању програма (у противном, она ће се једном сакрити и остаће скривена стално).

Програм за летилицу онда постаје:



Побољшање по питању евиденције броја живота. Скривајући летилицу током контакта, избегавамо могућност смањења броја живота за више од јединице. Летилица ће се затим поново појавити (пошто је астероид прешао свој пут).

Педагошка напомена

Овде, летилица, током судара, нестаје за тренутак. Програм ће бити побољшан касније ([Час 7](#)), када ћемо искористити костиме и поруке које ће омогућити промену изгледа летилице успешније него њено скривање, односно током судара нестаће астероид.

Заједничко представљање

Заједничко представљање ће пружити могућност размене потешкоћа с којим су се суочиле групе, и дати им времена да модификују своје програме у светлу различитих искустава. Ученици су такође разматрали функционалности које је требало програмирати.

Закључак

Наставник усмерава ученике ка формулисању неког закључка:

- било који комплексан задатак, који се може аутоматски решити, може да се растави на скуп једноставних задатака.
- једноставни задаци које смо данас програмирали садрже:
 - о промену варијабли: нека варијабла је простор у меморији у ком се може стокирати нека вредност (која јој се приписује). Ту вредност можемо користити у тестовима или каснијим прорачунима, али и извршити њену модификацију у сваком тренутку.
 - о петље: ове инструкције омогућују вишеструко понављање у блоку инструкција.
 - о тестови, којим се верификује да ли је неки услов испуњен или не.
- Ове инструкције су биле организоване у секвенце инструкција, при чему је свака покренута једним догађајем (клик на зелену заставицу).

Осим тога, ученици бележе, у своју свеску пројекта, добре праксе програмирања које су поменуте током овог часа.

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