

1, 2, 3, кодирај! – Активности циклуса 4 - Пројект « Програмирање видео игре на платформи » - Етапа 8: Регулисање одскока

Доминантна дисциплина	Математика
Резиме	Ученици програмирају начин одскока lika « Играча » од вертикалне ивице платформи. Затим се креирају функције за свако од понашања које је већ програмирано.
Појмови	« Језик » <ul style="list-style-type: none">• Језик <i>Скрач</i> програмирања је вођен догађајима. Слање порука је један од типова коришћених догађаја. « Добра пракса програмирања » : <ul style="list-style-type: none">• Постављање коментара у програму олакшава његово читање и размену с другим програмерима.• Добро написан програм, с добро дефинисаним именима варијабли и функција, не захтева много коментара.
Материјал	За наставника: <ul style="list-style-type: none">• Фајл <i>Скрач</i> Platformer_V07_demo.

Полазна ситуација

Наставник покреће програм Platformer_V07_demo, који одговара оном што је урађено на часовима 1 до 6, с пејсажом елемената платформе која формира зид. Указује на очигледне дисфункционалности које ученици нису опазили:

- Кад лик « Играч » скочи и додирне платформу вертикалном ивицом, он се врати на платформу и пење се док се не постави стабилно на њу.
- Кад лик « Играч » наиђе на плочицу са стране он ће је прећи.

У питању је програмирање следеће функционалности из менталне карте:

- Аватар: одскаче од зидова.
- Стандардни елемент платформе је непробојан за аватара (са стране)

Ученици морају да реше ове проблеме, али претходно заједнички праве лист с мапом програмирања.

Регулисање одскока од вертикалних платформи (заједнички а затим и у пару)

Наставник предлаже ученицима да размисле о начину на који ће регулисати одскок lika « Играч » од вертикалних ивица платформи. Саветује им да анализирају како су програмирали одмор lika « Играч » на платформама и њиховог одскока од плафона, и да се затим запитају да ли, у овој ситуацији, могу да прибегну истој процедури за вертикалне ивице платформе. После неколико минута размишљања које су имали ученици на располагању организује заједничко представљање идеја.

Неке групе ученика ће сигурно запазити да се одмор на платформама, и одскок од плафона добијају манипулацијом варијабле « вертикална брзина ». Ако то није случај онда наставник усмерава одељење ка овом запажању:

- да би скочио даје почетну вредност, 7, за варијаблу « вертикална брзина »:

нека вертикална брзина буде 7

- за мировање на платформи даје вредност 0.
- за одскок од плафона мења знак ове вредности:

нека вертикална брзина буде 0 - гравитација

- за убрзање наниже додаје се вредност варијабле гравитација (негативан знак је последица одабране конвенције да се узима позитивна вредност за гравитацију док у случају када је Играч усмерен на ниже долази до смањења његове ординате што се реализује помоћу « вертикалне брзине »:

нека вертикална брзина буде вертикална брзина - гравитација






Затим се одређује позиција ординатних оса помоћу инструкције:


промени у за вертикална брзина

За сад не користимо варијаблу « хоризонтална брзина » јер се померања десно/лево добијају модификацијом вредности апсцисе « x » lika « Играч ». Разлог за то је што су путање lika нереалне при скоковима. Наиме, ако престанемо да притискамо десну стрелицу на тастатури, током реализације скока, лик нагло престаје да

се креће десно/лево, а може и да промени правац када је у пуном скоку... као што се то дешава на цртанм филмовима (то је био и случај игре на платформи коју смо показали на часу 1), што ипак није могуће у реалности!

Наставник охрабрује ученике да уведу нову варијаблу « хоризонтална брзина », коју ће користити за вођење померања десно/лево (што подразумева модификацију постојећег подпрограма за лик « Играча ») затим то користи и за вођење одскока од вертикалних ивица. Све ово може да се уклопи у следећу листу пута програмирања која се конструише заједнички (изузев задатак 6 који ученици сигурно нису у стању да предвиде):

Функционалности програма	Природа задатак које је потребно решити	Тежина
1 – лик « Играч » се помера десно/лево ако притиснемо стрелице десно/лево	Задатак 1: креирати варијаблу « хоризонтална брзина » доступну свим ликовима. Нека почетна вредност ове варијабле буде 0.	
	Задатак 2: модификовати подпрограм « померање десно/лево » лика « Играч » на такав начин да се померања десно/лево добијају фиксирајући прво вредност варијабле « хоризонтална брзина », а затим и додавајући « хоризонталну варијаблу » варијабли « x ».	
	Задатак 3: (факултативно): реализовати на неки начин ситуацију у којој стрелице десно/лево делују само кад лик « Играч » додирне неку платформу одозго.	
	Задатак 4: (факултативно): да би се узело у обзир трење на подлози омогућите лику « Играч » да постепено успорава своје померање десно/лево кад престанемо да притискамо стрелице десно/лево, а када се он поставља на неку платформу.	
	Задатак 5: променити знак хоризонталне брзине када лик наиђе на вертикалну ивицу платформе, да би се на тај начин произвео ефекат одскока.	

2 – лик « Игач » одскаче од вертикалних ивица платформи	Задатак 6: спречити могућност да лик дође истовремено у контакт с горњим делом платформе и њеном ивицом (случај кад постављене плочице формирају зид).	
---	--	---

Ученици, пошто креирају копију фајла V07 с последњег, почињу самостално да реализују задатке дате у претходној листи, док наставник иде од групе до групе и дискутује с ученицима о напредку који су остварили током досадашњег рада. Подсећа ученике да редовно меморишу свој пројект.

Следе елементи корекције за задатке 1 до 6:



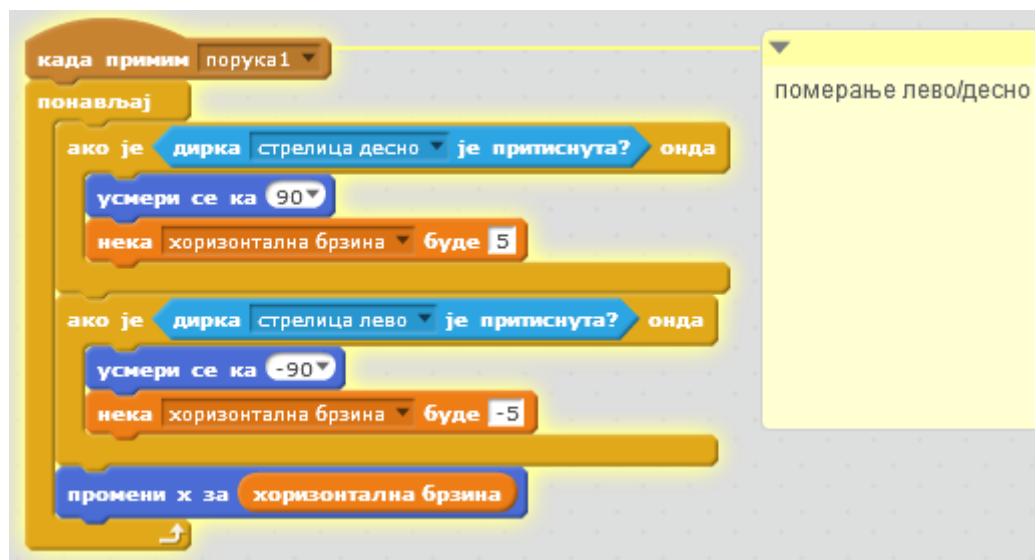
Задатак 1: креирати и дати почетну вредност варијабли « хоризонтална брзина »

Давање почетних вредност некој променљивој је описано у [Етапи 2](#) (корекција задатака 1 и 3). Смислено је поставити варијаблу « хоризонтална брзина » на истом месту с варијаблом « вертикална брзина », тј., у подпрограм за давање почетних вредности лика « Игач ».



Задатак 2: водити хоризонтално померање Игача

Користимо овако креирану варијаблу « хоризонтална брзина » при командовању померања десно/лево лика « Игач ». Модификације подпрограма написане на [часу 1](#) су заиста минималне:

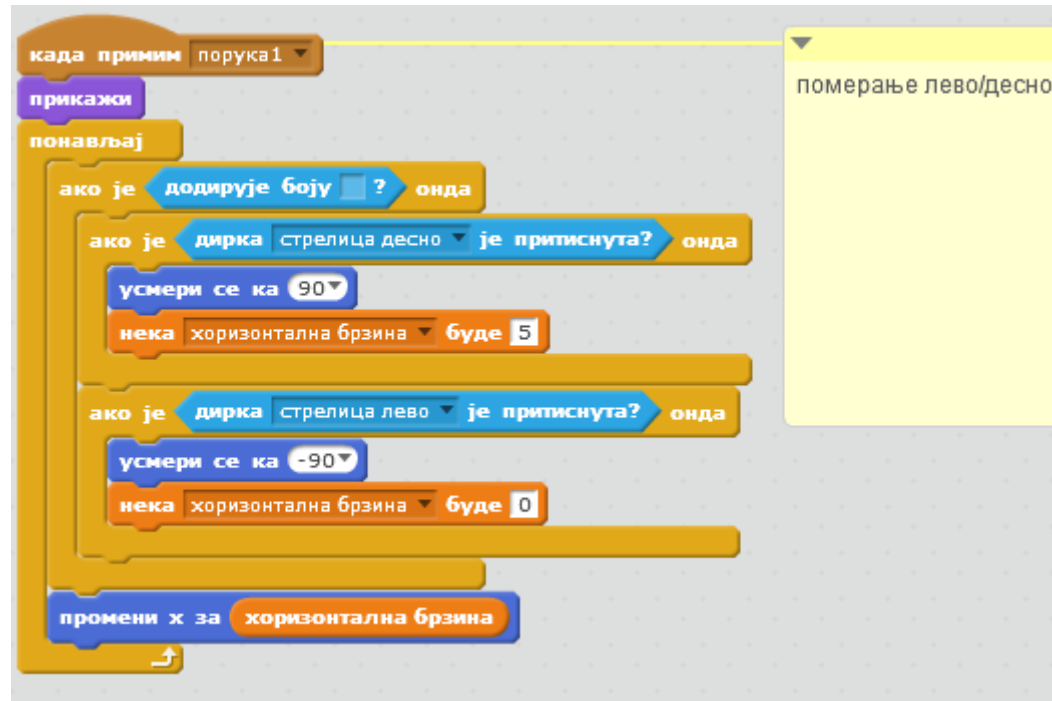


Ова кетања, у овом тренутку, могу бити вођена било да је лик « у ваздуху » било да је на некој платформи. Осим тога, његово кретање не престаје чак и када смо престали да притискамо стрелице лево/десно на тастатури.



Задатак 3 (факултативно): избећи пормену правца током скокова

Када се лик « Играч » одмора на горњој површини платформи, уствари додирује специфичну зелену боју на њиховим горњим површинама. За решење задатка 3, можемо да додамо овај услов контакта у подпрограму који контролише померања лево/десно за лик « Играч » :



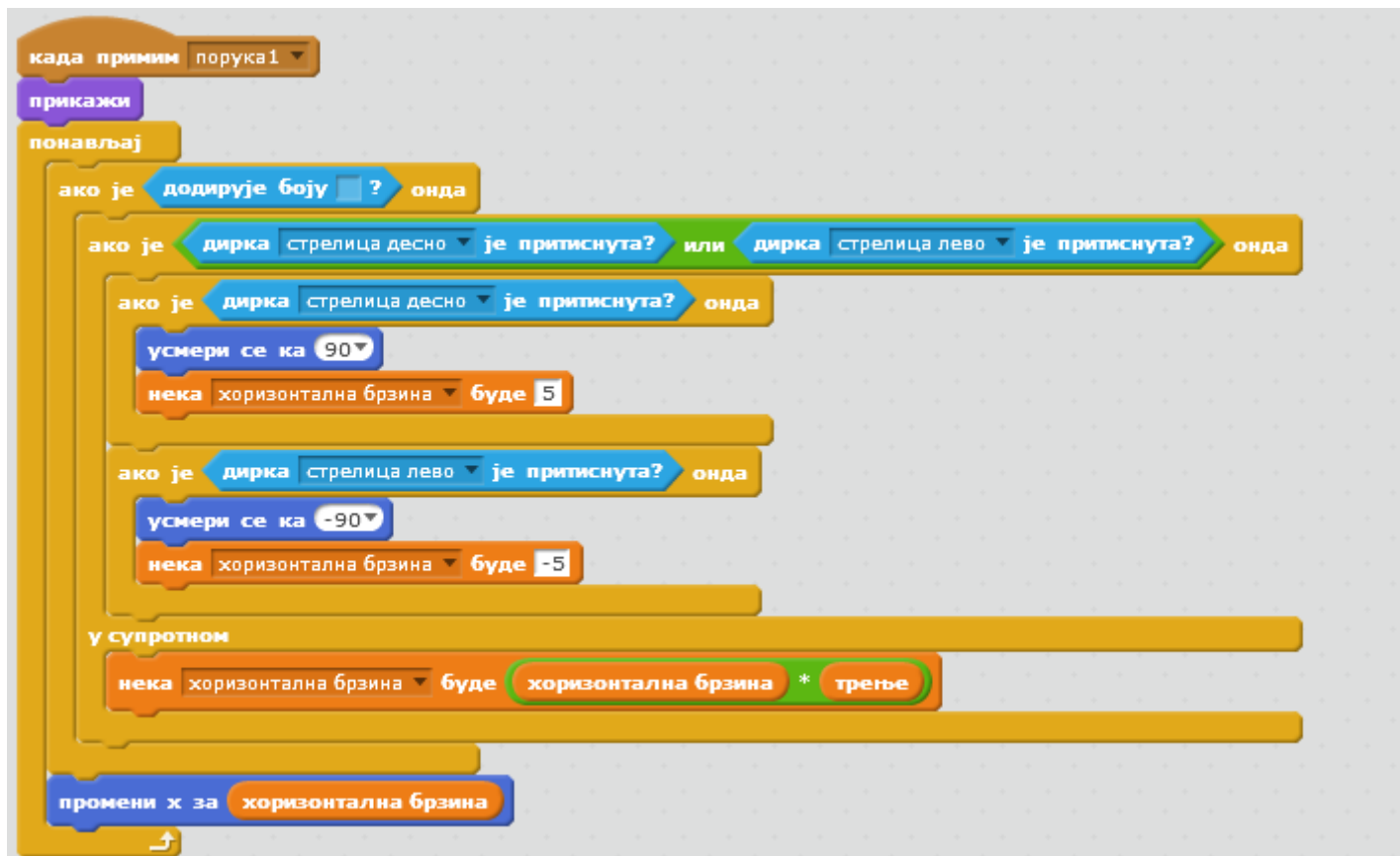
Ажурирање позиције лика у односу на апсисне осе мора бити реализовано на основу услова истинитости или лажности контакта. Инструкција « додај хоризонталну брзину на x » не мора да зависи од овог услова па га постављамо ван условне петље « ако ... онда ».



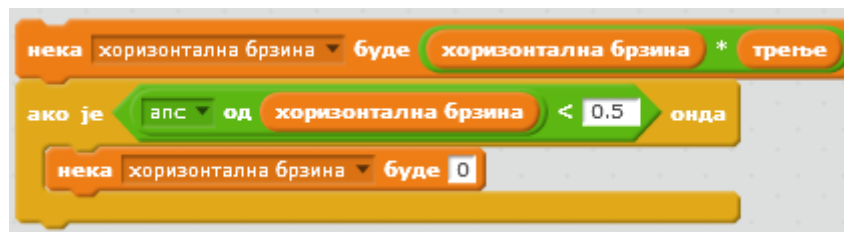
Задатак 4 (факултативно): узети у обзир трење на тлу

Решење задатка 4 претпоставља да је вредност променљиве « хоризонтална брзина » нула ако лик « Играч » додирне платформу одозго (томе одговара зелена боја) и да стрелице десно/лево нис у притиснуте. Да би се добио резултат можемо да množимо вредност брзине са « коефицијентом трења » мањим од 1, што претпоставља унапред креирање варијабле « трење » и давање почетне вредности (на пример на вредност 0.5).

Морамо да делујемо посредством условних инструкција « ако ... онда » и « ако ... онда ... аконе », што је с концептуалног аспекта тешко. Следећи предлог није најкраћи могућ, али је најједноставнији за разумевање:

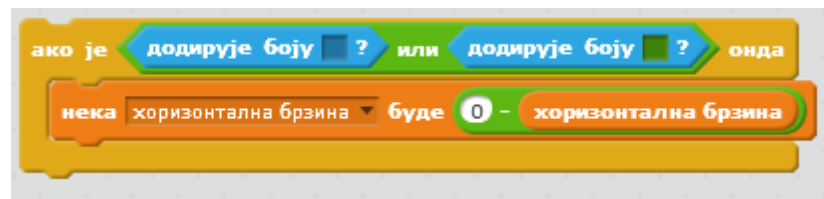


Ако се на сцени прикаже вредност варијабле « Хоризонтална брзина », онда се уверавамо да ће за лик у фази успоравања ова варијабла узимати врло мале вредности попут 0,00000 Што није толико лоше јер се ова вредност приближава 0 као у овом примеру:

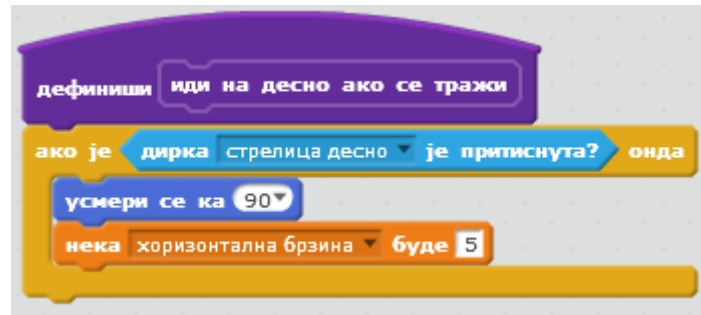


Задатак 5: одскок од зидова

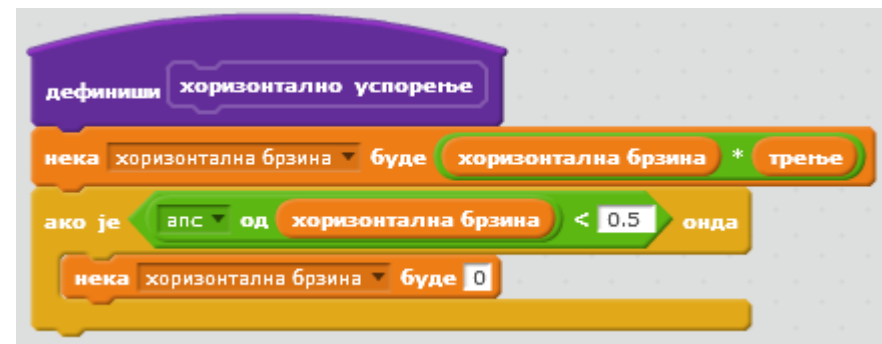
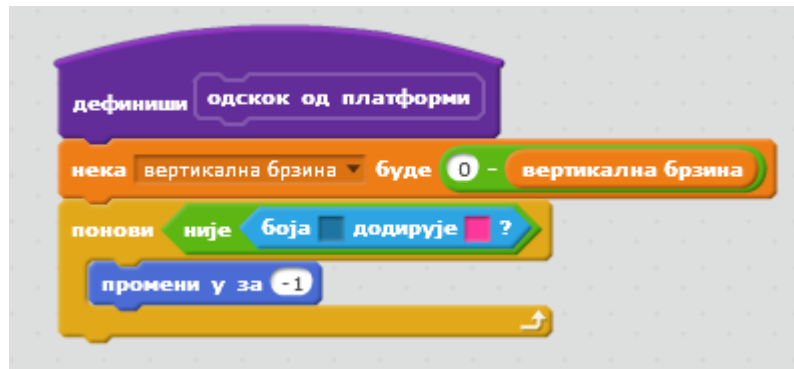
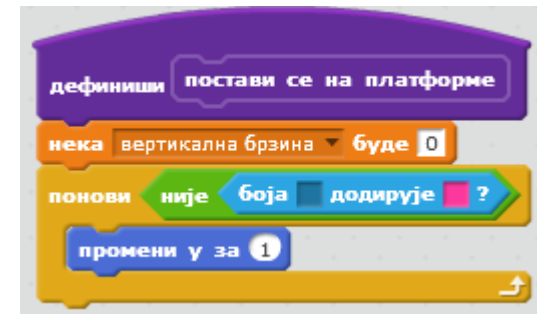
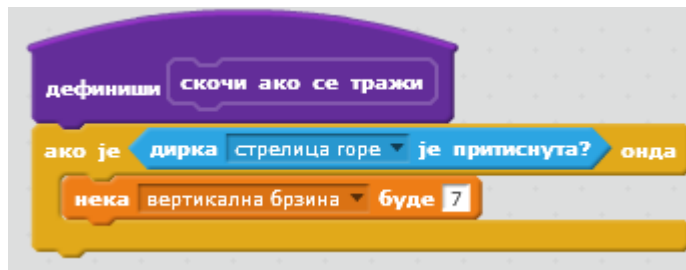
За програмирање овог задатка морамо да тестирамо контакт лика с бојом вертикалних ивица плочица и променимо знак хоризонталне брзине само ако је до контакта дошло, на пример:



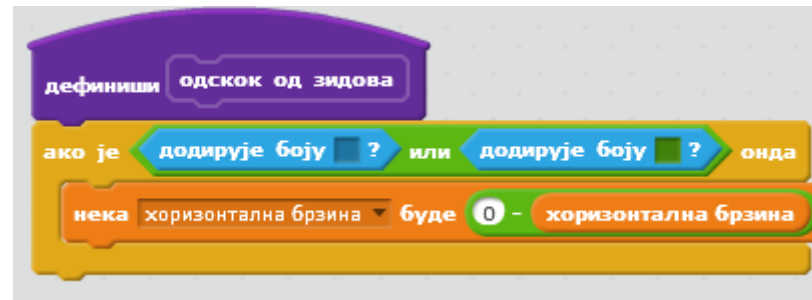
Сад се поставља питање где поставити ове инструкције у програму за лик « Игравч »? Већина начина тестирања ученика у вези овог проблема је дала незадовољавајуће резултате, јер постоји конфликт између контакта с палтформом који могу бити на/под/бочно. Осим тога, није тако једноставно реорганизовати подпрограме с циљем да се тестирају ефекти различитих веза. Ученици ће се једноставно у тим покушајима изгубити. Време је да се креирају функције које одговарају сваком од ових блокова на начин како је то урађено у Стапи 4. Тако можемо, без модификације функција, променити начин њиховог позивања, и одабрати опцију која нам најбоље одговара за оно што желимо да нађемо. На тај начин можемо креирати две функције за померање на десно и на лево (функција која омогућује померања на десно је приказана овде, а ова друга се конструише на исти начин):



Можемо креирати и функцију за контролисање скокова, потом за постављање на платформе, затим за одскок од плафона и за постепено успоравање лика:



Коначно и функцију која се намеће за одскок од зидова:



Остаје нам само да нађемо како да ефикасно уредимо позивање ових функција. Боље је избрисати подпрограме који су до сад регулисали померање десно/лево, као и хоризонтално и вертикално кретање, и почети од нуле како се то каже у енглеском! Следећи програм функционише боље, али то и није једина могућност:

```

када прими повика1
иди на x: 100 y: 150
прикажи
понављај
  промени x за хоризонтална брзина
  одскок од зидова
  промени y за вертикална брзина
  ако је додирујебоју ? онда
    постави се на платформи
    скочи ако се тражи
    ако је дирика стелила десно је притиснута? или дирика стелила лево је притиснута? онда
      иди на лево ако се тражи
      иди на десно ако се тражи
    хоризонтално успорење
  у супротном
    ако је додирујебоју ? онда
      одскок од платформи
    у супротном
      нека вертикална брзина буде вертикална брзина - гравитација

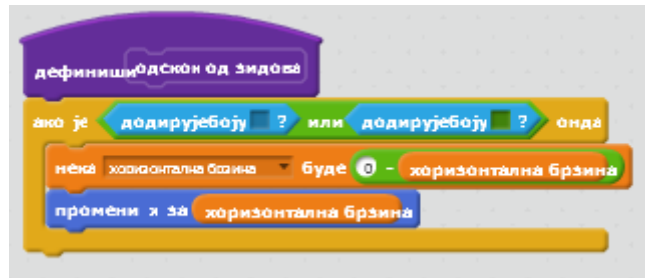
```



Задатак 6: побољшати непробојност зидова

У сваком случају проблем је још увек велик, јер лик може да прође зидове, ако се креће бочно по платформама. То је потпуно нормално ако детаљно анализирамо програм! Јер програм обично уради оно што од њега захтевамо ...

Следећа модификација одскока од зидова омогућује да се узме у обзир и њихова реална непробојност:



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