



"Ss. Cyril and Methodius" University in Skopje  
**FACULTY OF COMPUTER  
SCIENCE AND ENGINEERING**

Проектна задача по предметот Програмирање на  
видео игри

## **Track & Wheels: Formula Racing**

---

**Професор:**

Д-р Катарина Тројачанец

**Изработиле:**

Сара Гроздановска 221103

Моника Наумовска 223118

## Содржина

1. Вовед .....	3
2. Имплементација .....	4
Почетно мени .....	4
Мени за избор на режим .....	5
Ниво .....	6
Ноќен ефект .....	8
Leaderboard .....	9

# 1. Вовед

„Track & Wheels: Formula Racing“ е возбудлива и динамична тркачка игра изградена со библиотеката Pygame. Играта е дизајнирана да понуди адреналинско искуство каде играчот управува со свој автомобил и се соочува со различни предизвици тесни кривини, случајно поставени пречки.

Она што ја прави играта особено интересна е можноста играчот сам да избере како ќе игра да вози самостојно и да ги тестира своите вештини, да се натпреварува со пријател на истата тастатура, или да се соочи со вештачка интелигенција која управува со компјутерска кола. Секој од овие режими нуди различно искуство и го прави играњето повторно возбудливо, дури и по повеќе обиди.

Посебна внимателност е посветена на патеките. Тие не се едноставни форми, туку се графички изработени во GIMP, што овозможува креативност и уникатен изглед за секое ниво. Патеките се различни по дизајн и распоред, со што играчот секогаш добива нов предизвик. Дополнителна динамика носат пречките кои се појавуваат на случајни позиции, правејќи секое повторно одигрување уникатно.

Играта има и визуелен елемент кој додава дополнителна возбуда екранот постепено потемнува за време на трката, создавајќи впечаток на ноќна атмосфера. Овој ефект не само што изгледа привлечно, туку и го зголемува предизвикот.

По завршувањето на сите нивоа, постигнатите резултати вкупното време, просечната брзина и бројот на победи се запишуваат во лидерборд, каде што играчите можат да се споредат меѓу себе. Овој систем додава натпреварувачки дух и мотивира повторно играње со цел подобрување на резултатите.

## 2. Имплементација

### Почетно мени

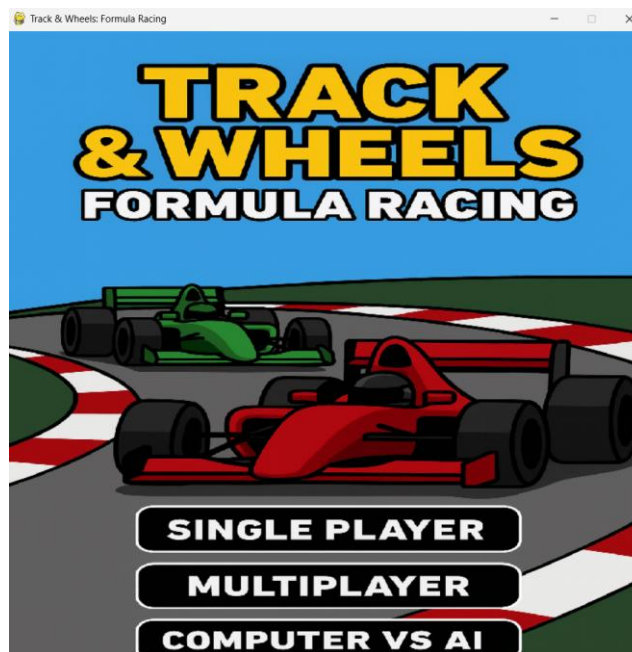


Слика1: Почетно мени

#### **show\_menu()**

Методот прикажува почетно мени со позадина и копче за старт. Играта останува во циклус сè додека играчот не кликне на копчето или не го затвори прозорецот. Тоа е првиот екран што го гледа секој играч.

## Мени за избор на режим



Слика2:Менито за избор на режим

### **show\_mode\_menu()**

Овој метод го прикажува менито за избор на режим. Играчот може да избере еден од трите режими: самостоен , мултиплеер или против компјутер (AI). Изборот се прави со клик на копче, а методот враќа кој режим е избран.

### **PlayerCar.reduce\_speed()**

Метод кој постепено ја намалува брзината на автомобилот кога играчот не притиска тастери. Со ова возењето изгледа пореално, бидејќи автомобилот не застанува веднаш.

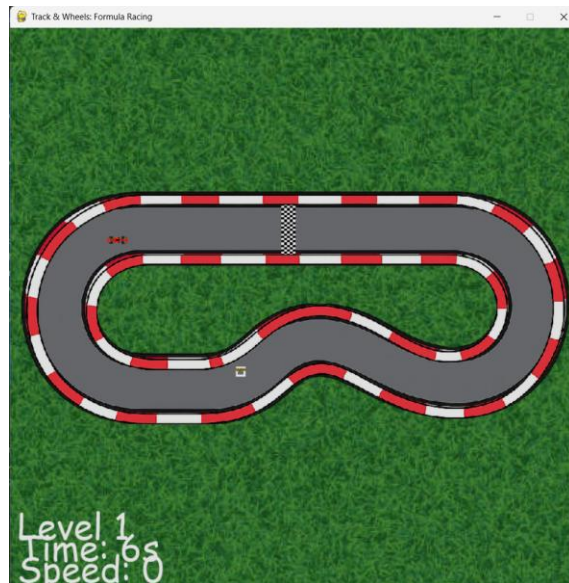
### **PlayerCar.bounce()**

Кога автомобилот ќе удри во сид или пречка, овој метод ја превртува брзината и го движи автомобилот назад. Тоа создава ефект на „одбивање“ и прави играта поинтересна.

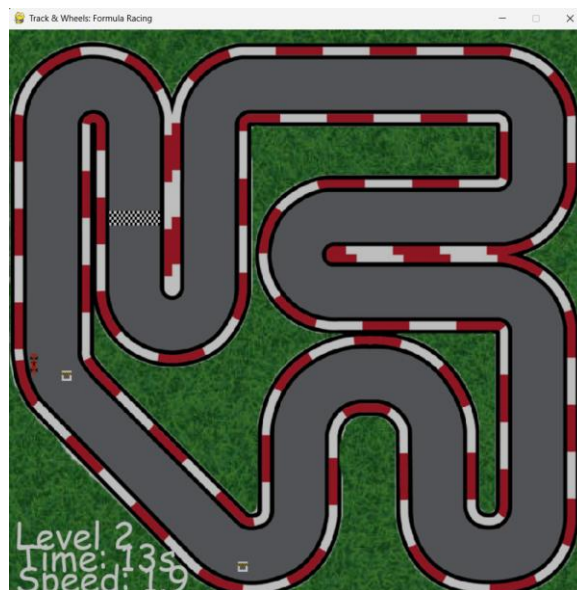
### **ComputerCar.calculate\_angle()**

Метод кој ја пресметува насоката на компјутерската кола според следната точка на патеката. Автомобилот автоматски го врти аголот за да ја следи патеката. Ова овозможува AI противникот да се движи „паметно“.

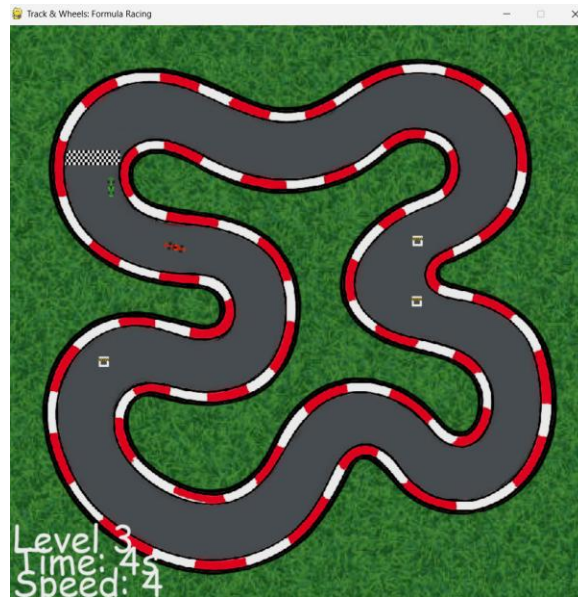
## Ниво



Слика3:Ниво 1



Слика4:Ниво 2



Слика5:Ниво 3

### **load\_level(level\_num)**

Овој метод вчитува ново ниво според бројот. Ја поставува патеката, стартните позиции, финиш линијата и случајните пречки. Со тоа секое ниво е различно и носи нов предизвик.

### **ComputerCar.update\_path\_point()**

Овој метод проверува дали компјутерската кола стигнала до тековната целна точка на патеката. Ако стигнала, преминува на следната точка. Со тоа, колата постојано напредува низ патот.

### **ComputerCar.move(obstacles=None)**

Главниот метод за движење на компјутерската кола. Најпрво проверува дали има пречки и се обидува да ги избегне. Ако нема пречки, автомобилот нормално ја следи патеката. Со ова, AI изгледа како вистински противник што внимава на патот.

### **ComputerCar.will\_hit\_obstacle(obstacles)**

Метод што предвидува дали компјутерската кола ќе удри во пречка ако продолжи по тековната насока. Пресметува идната позиција и проверува дали таму има пречка. Ова му овозможува на AI да реагира навреме.

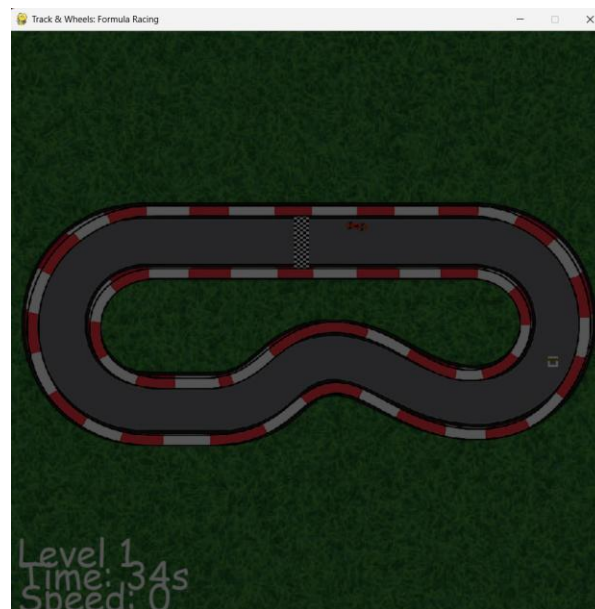
### **ComputerCar.bounce()**

Исто како кај играчот, и кај компјутерската кола овој метод ја менува брзината и ја движи наназад по удар. Со тоа, компјутерот реално реагира на судири.

### **ComputerCar.next\_level(level, start\_pos)**

Овој метод ја подготвува компјутерската кола за следно ниво. Ја поставува на стартна позиција и ја зголемува брзината за малку, што прави натпреварот потежок како што играта напредува.

## Ноќен ефект



Слика6: Ноќен ефект

### **draw(win, images, player\_car, second\_car, game\_info, obstacles=None)**

Главниот метод за цртање на екранот. Тој прикажува слики од позадината, автомобилите, пречките и информациите (брзина, време, ниво). Дополнително додава „ноќен ефект“ со постепено затемнување.



## Leaderboard



Place	Player	Avg Speed	Time (s)
1st	Player	0.5	8.3
2nd	Player	0.6	10.1
3rd	Player	2.4	24.7
4th	Player	0.0	39.2
5th	Player	1.7	89.6
6th	Player	2.0	137.6
7th	Player	0.0	27.5

Press SPACE to play again or ESC to exit

Слика7: Leaderboard

**show\_leaderboard(win, leaderboard, game\_stats, player\_name="Player")**

Метод што го прикажува екранот со резултати. Го зачувува резултатот на тековниот играч, ги прикажува најдобрите и дава можност за повторно играње или излез.

**move\_players(player1, player2=None)**

Методот ги следи тастерите кои ги притискаат играчите. Првиот играч користи WASD, вториот стрелки. Ако никој не притиска ништо, автомобилите постепено ја намалуваат брзината.

**handle\_collision\_all(cars, track\_border\_mask, finish\_mask, finish\_position, start\_pos, game\_info, mode, obstacles=None)**

Метод кој проверува дали автомобилите удриле во сид, во пречка или ја поминале финиш линијата. При удар се пушта звук на судир и автомобилот застанува или се одбива назад. Ако се помине финишот, играчот победува и се преминува на следното ниво.

## **main\_game()**

Ова е централниот метод кој ја стартува и контролира целата игра. На почеток прикажува мени, овозможува избор на режим и ги подготвува автомобилите и патеките. Потоа постојано ја освежува графиката, проверува движења и судири и го следи напредокот низ нивоата. По завршување на сите нивоа, се прикажува лидербордот.

## **Класа AbstractCar**

Класата AbstractCar е основа за сите формули во играта. Во неа се дефинираат брзината, аголот, позицијата и забрзувањето. Автомобилот може да се врти (rotate), да тргне од почетна позиција (set\_start\_pos) и да се нацрта на екранот (draw). Со методите move\_forward и move\_backward се менува брзината, а move ја пресметува новата позиција според аголот. Судирите се проверуваат со collide и collide\_with\_obstacle, а со reset автомобилот се враќа на старт.

## **Класа GameInfo**

Класата GameInfo служи за следење на целата состојба и статистика во играта. Таа чува на кое ниво е играчот, дали е започнато ниво, времето на старт, бројот на победи и дали играта е завршена. Со методите се менаџираат победи (add\_win), премин на следно ниво (next\_level), ресетирање (reset), и проверка дали играта е готова (game\_finished). Исто така, се следи времето по ниво (get\_level\_time), вкупното време на играта (get\_total\_time), брзините на играчот (add\_speed) и просечната брзина (get\_average\_speed). На крај, со get\_stats се добиваат сите статистики заедно.