

Програмирање на видеоигри

Лабораториска вежба 2

Space Scavenger

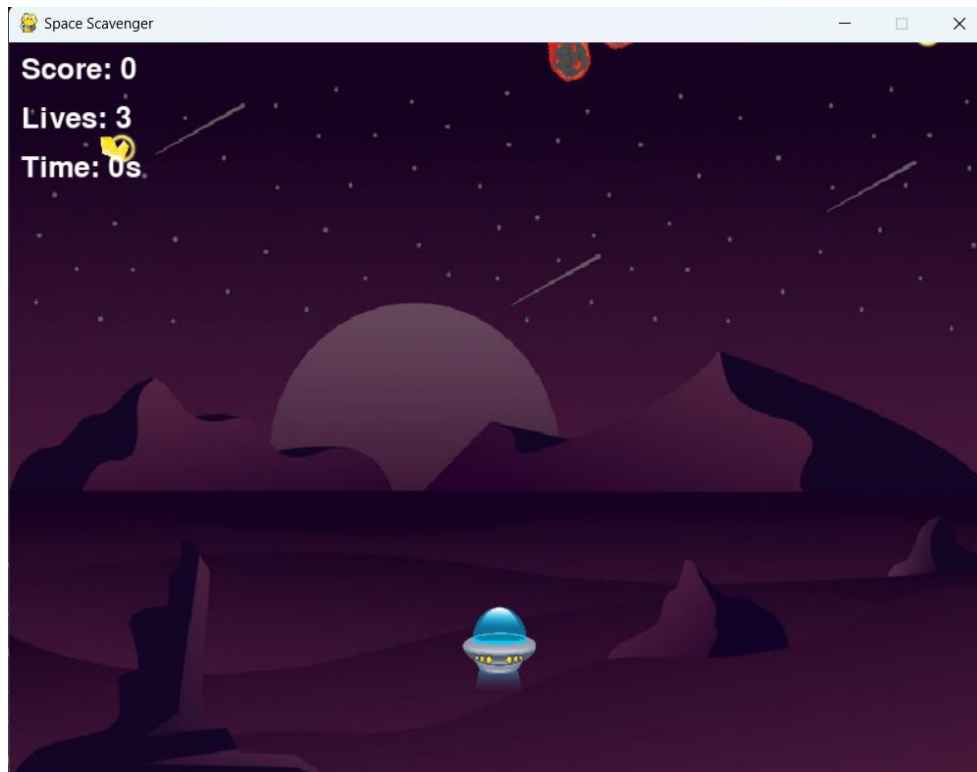
Дарко Димитров 213083

1. Вовед

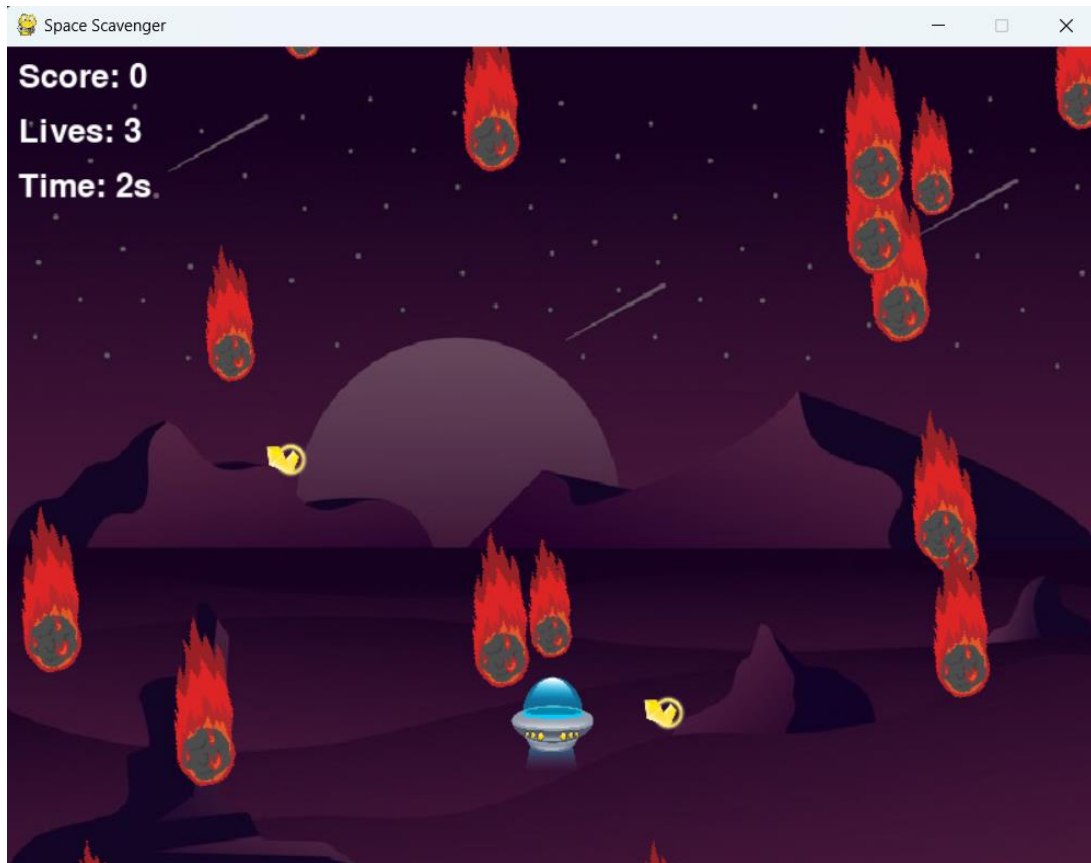
"Space Scavenger" е динамична игра развиена со *pygame*, каде играчот управува со вселенски брод, избегнува астероиди и собира енергетски кристали за да го зголеми резултатот. Астероидите стануваат побрзи и поголеми со текот на времето, додека кристалите му даваат поени и нови животи на играчот. Играта ја збогатуваат звучни ефекти и позадинска музика, создавајќи возбудливо и интерактивно искуство.

2. Почетен екран

На почетниот екран се прикажува позадинска слика која го поставува темпото и атмосферата на играта. Играта започнува автоматски по отворањето на апликацијата, со што вселенскиот брод, астероидите и кристалите се појавуваат на екранот, подготвувајќи го играчот за акцијата.



3. Тек на играње



Започнување на играта:

- Играта започнува со приказ на вградената позадина и вклучување на позадинска музика која ја создава авантуристичката атмосфера. Играчот управува со вселенски брод кој започнува во долниот дел од екранот. Течноста на играта е динамична и исполнета со предизвици каде што треба да се избегнуваат астероиди и да се собираат кристали.

Контроли и движење на играчот:

Играчот го контролира бродот со стрелките на тастатурата (лева и десна стрелка), што му дозволува да се движи по хоризонталната оска во рамките на екранот. Целта е да се избегнуваат астероидите, кои паѓаат од врвот на екранот. Движението е ограничено на просторот на екранот, така што бродот не може да излезе надвор од него.

Појава на астероиди и кристали:

Секој пат кога поминува време, нови астероиди се појавуваат и паѓаат од горниот дел на екранот. Нивната големина и брзина се случајни, што додава дополнителен предизвик за играчот. Понекогаш, на екранот се појавуваат и кристали кои треба да се соберат. Собирањето кристали ја зголемува заработената поени и понекогаш дава бонус животи.

Судри и последици:

Кога бродот ќе се судри со астероид, играчот губи живот. Во моментот на судир, се слуша звучен ефект што ја потврдува штетата. Ако играчот изгуби сите животи, играта завршува и на екранот се прикажуваат резултатите, времето што е поминато и опцијата да се започне играта одново.

Собирање кристали:

Кога бродот ќе се сретне со кристал, се прикажува звучен ефект на собирање и резултатот се зголемува. Ако играчот собере 10 кристали, добива дополнителен живот, што му овозможува да продолжи со играта подолгу и да ја зголеми својата издржливост.

4. Крај на играта

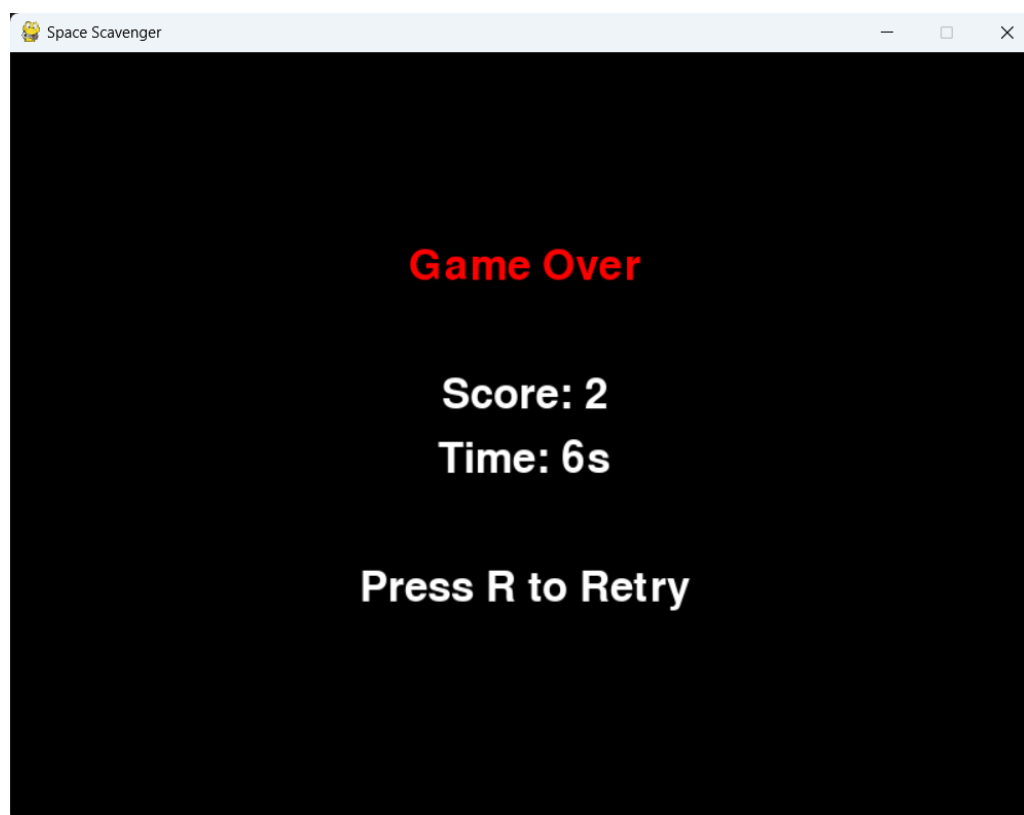
Играта завршува кога играчот ќе ги изгуби сите животи, по што се прикажува екран за пораз. На екранот за пораз се прикажуваат следниве информации:

- **„Game Over“** - Порака која го информира играчот дека ја изгубил играта.
- **Резултат** - Прикажување на вкупниот број на собрани кристали до моментот на поразот.
- **Време** - Количината на време поминато во играта до моментот на изгубени животи.

По прикажувањето на овие податоци, играчот има можност да ја започне играта повторно со притискање на тастатурата на копчето **R**. Ако играчот не сака да продолжи, може да ја затвори играта.

Додека се прикажува поразот, се активира звукот за судир за да ја нагласи моменталната ситуација, додавајќи уште еден ефект во играта.

Притоа, сите игри активности се запираат, сите астероиди и кристали исчезнуваат, а вселенскиот брод на играчот се враќа на почетната позиција. Потоа, играчот може да ја започне играта одново со нов број на животи.



5. Имплементација

Player (класата)

Го претставува играчот како вселенски брод.

Методи:

- `__init__()`: Го иницијализира бродот со слика, позиција, брзина и маска за детекција на судир.
- `update(keys)`: Го ажурира движењето на бродот базирано на внесените копчиња.
- Се користи за управување на играчот преку тастатурата.

Asteroid (класата)

Ги претставува астероидите во играта.

Методи:

- `__init__(game_time)`: Генерира астероид со случајна позиција, брзина и големина.
- `update()`: Ја ажурира позицијата и го отстранува астероидот ако излезе од екранот.
- Се користи за создавање и управување на непријателските објекти во играта.

Crystal (класата)

Ги претставува енергетските кристали кои играчот ги собира.

Методи:

- `__init__()`: Генерира кристал со случајна позиција и брзина.
- `update()`: Ја ажурира позицијата и го отстранува кристалот ако излезе од екранот.
- Се користи за додавање предизвици и награди за играчот.

game_over_screen(score, time_elapsed)

Задолжена за приказ на екранот „Game Over“ по завршување на играта. Таа прикажува текстуални информации како „Game Over“ како наслов, постигнатиот резултат (score) и времетраењето на играта (time_elapsed), како и инструкција за рестартирање со текстот „Press R to Retry“.

Корисничката интеракција е овозможена преку проверка на внесените копчиња, па доколку корисникот го притисне копчето R, играта се рестартира.

Дополнително, се проверува настанот за затворање на играта за да се прекине програмот доколку е потребно. Приказот на текстот е центриран на екранот, а петљата на оваа функција се извршува сè додека корисникот не преземе акција.

Главниот while loop

Ја контролира целокупната логика на играта и интеракциите. Во секој циклус се обновува позадината на екранот со прикажување на позадинската слика.

По случаен избор се создаваат нови астероиди и кристали, кои се додаваат во соодветните групи. Играчот се контролира преку внесови од тастатура, а неговата позиција се ажурира според движењата.

Циклусот ги обработува и колизиите: при судир со астероиди се намалува бројот на животи, а доколку се изгубат сите животи, се повикува game_over_screen.

При судир со кристали, резултатот се зголемува, а на секои 10 поени се додава еден живот. Исто така, се прикажуваат резултатот, времето и бројот на животи на екранот, кои се ажурираат во секој циклус. Времетраењето на играта (game_time) се зголемува за 1/60 секунда на секој фрејм, обезбедувајќи точна временска мера.

Циклусот исто така проверува за настани како што е QUIT за да овозможи излез од играта кога тоа е побарано. Со оваа структура, главната петља ги поврзува сите елементи на играта и обезбедува синхронизирано функционирање.