

Space Scavenger – Документација за Лабораториска Вежба 2

1. Објаснување на Играта

Space Scavenger е едноставна, но и предизвикувачка игра наменета за една личност. Во почетната состојба на играта, на екранот е само вселенското летало, коешто го раководи играчот. Играта ја започнувате со 3 животи и 0 поени, а целта на играта е да добиете 1000 поени пред да ги искористите сите 3 животи.

Кога започнува играта, од двете горни кошеви паѓаат метеори, а низ екранот се поставени разни кристали коишто леталото треба да ги собере. Со секој кристал што ќе го собере играчот, вселенското летало станува побрзо, што дозволува полесно да се избегнуваат метеорите, но и се зголемува бројот на поените (10 поени за секој кристал).

Ако играчот се судри со метеор, губи 1 живот, и со секој живот изгубен, метеорите стануваат побрзи и поголеми. Дополнително, со секои 100 поени добиени (10 кристали), метеорите стануваат дополнително побрзи и поголеми.

Тежината на играта е во главно концентрирана околу втората половина, односно преживување до 1000 поени (лично само еднашка успеав во 30тина обиди 😊)

2. Објаснување на Кодот

- **Глобални променливи, листи, позадини, ресурси**

На почетокот се декларираат и иницијализираат сите неопходни елементи за понатамошно работење и градење на играта. Се поставува големина на прозорецот, импортираат сите потребни библиотеки како и сите аудиовизуелни елементи за играта и за крај се декларираат и иницијализираат некои глобални променливи и низи коишто ќе се користат понатаму (пр. Score, lives, итн...)

- **create_asteroid, create_crystal**

Овие две функции ги генерираат метеорите и кристалите соодветно.

Кристалиите се релативно едноставни имплементирани, со тоа што се генерираат во горниот дел од екранот во случајна координата и паѓаат надолу.

Метеорите пак, се генерираат случајно или во горелевиот или горедесниот кош (еднаква шанса), и паѓаат дијагонално надолу. Насоката зависи од тоа од кој кош паѓаат, а за брзината е дадена да биде различна за сите (од даден опсег). Исто така, тука се имплементирани идните зголемувања на метеорите, којшто зависат и се множат со `asteroid_scale_factor`, којшто е поставен на 1.

- **draw_objects, draw_score, draw_lives**

Како што имплицира и самото име на функциите, тука се цртаат објектите на играта (позадината, вселенското летало, метеорите и кристалите), но и резултатот на играчот којшто се прикажува горелево, но и бројот на животи којшто го има којшто се прикажува горедесно.

За позадината, метеорите и кристалите, се рендерираат дадените ресурси од материјалите, со тоа што, бидејќи во мојата игра има можност метеорите да паѓаат и од лева страна, имплементирам flip метод за да делува визуелно подобро кога паѓаат од левиот кош.

За животите и резултатот, прво се иницијализира фонот и големината, коешто потоа се рендерира како слика и прикажува на точна позиција на екранот.

- **respawn_spaceship**

Оваа функција постои поради системот на животи во играта. Секој пат кога ќе се судри леталото со метеор, се повикува оваа функција и прво проверува дали бројот на животи е поголем од 1. Доколку е, одзема еден живот, ги брише сите метеори на екранот и го враќа леталото во првобитна положба. Доколку бројот на животи е 0, се одзема последниот и се повикува game_over функцијата.

- **game loop**

Се воведува музиката на играта, којашто се повторува во бесконечен циклус. Потоа се декларираат можните движења на леталото (горе, доле, лево, десно) соодветно со копчињата на тастатурата.

Следно се разгледуваат движењата на метеорите. Односно доколку заминат од видното поле на прозорецот, се бришат. Дополнително, кога се случува колизија на бродот со метеор, се повикува respawn_spaceship функцијата.

Понатаму се раководи со кристалите, односно доколку заминат од видното поле на прозорецот се бришат, слично како и метеорите, но и доколку леталото ги собере (се судрат), се бришат од полето и се зголемува резултатот за 10.

Конечно, се воведени услови за зголемување на тежината (за секои 100 поени добиени), се создаваат наредни метеори и кристали и се додава услов за успех на играта и повикува соодветната функција.

- **game_over and success_screen**

Се повикуваат кога ќе се изгубат сите животи или соберат 1000 поени соодветни. Самиот изглед на двата прозорци не е нешто различен, ниту нивната имплементација. Се прикажува текст за соодветната крајна состојба на играта, како и постигнатиот резултатот одма под тоа. Потоа има мала пауза за да може играчот да го види резултатот, и програмата прекинува, заедно со музиката.

Изработено:

- Матеј Филиповски, 213271