



3D ИГРА ВО GODOT

Станува збор за едноставна 3D игра во која играчот има за цел да се движи низ парк и да собира богатства. Откако ќе се соберат сите богатства, во случајот кристали, играчот победува. Доколку времето истече и играчот не успеал да ги собере сите богатства или играчот паднал од површината на паркот, играта завршува.

Начин на играње:

- 1) Движење со помош на WASD копчињата од тастатурата.
- 2) Скокање на Space
- 3) При интеракција со кристал, се зголемува резултатот на играчот и го снемјува кристалот.
- 4) Сите кристали собрани носат до победа, во секој друг спротивен случај играчот губи.

Главни функционалности и имплементација

1. Имплементација на паркот

За имплементирање на 3Д околината го користев ProtonScatter plugin кој овозможува на многу едноставен и брз начин да креирам парк со дрвја, грмушки и слично. Содржи множество од 3д објекти кои можат да се користат. Исто така за карактерот и текстурата на тревата користев веќе креирани објекти, преземени од [Polyhaven.com](https://polyhaven.com) и [Kenney.nl](https://kenney.nl).

- Подот е создаден со CSBox3D со CollisionShape3D за детекција на играчот и останатите објекти. При тоа додадена е и текстура на трева за пореалистичен ефект.
- Со DirectionalLight3D и WorldEnvironment се позиционира сонце во светот и небо, со кои се подобрува визуелниот изглед и се поставува светлината.

2. Движење на играчот

Движењето на играчот е имплементирано со користење на CharacterBody3D

- Со velocity и [move_and_slide\(\)](#) методи за манипулација со движењето.
- Додавање на гравитација за природно паѓање на играчот.
- Пресметка на влез од корисникот за движење напред, назад, лево и десно.

- Скокот е имплементиран преку проверка дали играчот е на подот и со користење на `jump_force`.

3. Собирање на богатствата (кристалите)

- Предметите се направени со `Area3D` и `CollisionShape3D` за детекција на нивната позиција во играта.
- При влез на играчот во просторот дефиниран со `Area3D` за тој кристал, се активира `body_entered` сигналот и кристалот се брише.
- При секое собирање на нов кристал, се зголемува резултатот и соодветно се ажурира на екранот.
- Кога сите кристали ќе се соберат, играчот победува.

2. Тајмер за играта

- Имплементиран е `Timer` во сцената, кој при стартување на играта, започнува да брои наназад, почнувајќи од 50.
- Кога времето ќе истече, ако не се собрани сите богатства, играта завршува како неуспешна.
- При победа, тајмерот се крие за да не се прикажува повеќе, а се прикажува порака за победа.

3. Камера за следење на играчот

Со `CameraPivot Node` кој содржи `Camera3D` се следи движењето на играчот, со тоа што позицијата на `CameraPivot` се синхронизира со таа на играчот. Со помош на стрелките од тастатура може да се ротира камерата околу играчот.

4. Детекција на пад од платформата

Со помош на `global_transform.origin.y` се следи позицијата на играчот во паркот. Ако позицијата падне под одредена граница (`fall_limit`), играта завршува како неуспешна, се прикажува порака за крај на играта.

5. Колизија со објекти

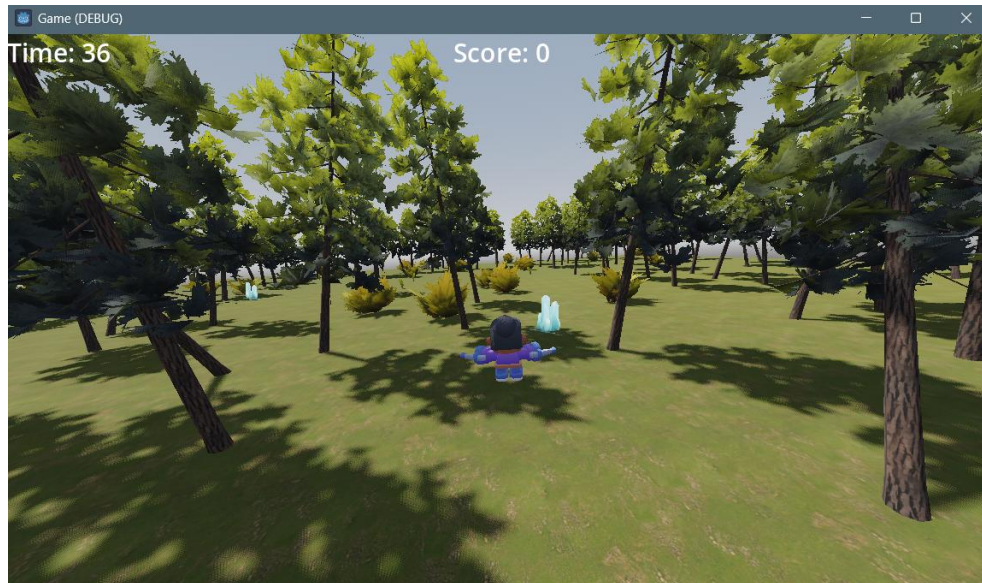
Со додавање на `StaticBody3D` во комбинација со `CollisionShape3D` на дрвјата и другите користени објекти се спречува играчот да поминува низ нив.

6. Графички елементи и кориснички интерфејс (UI)

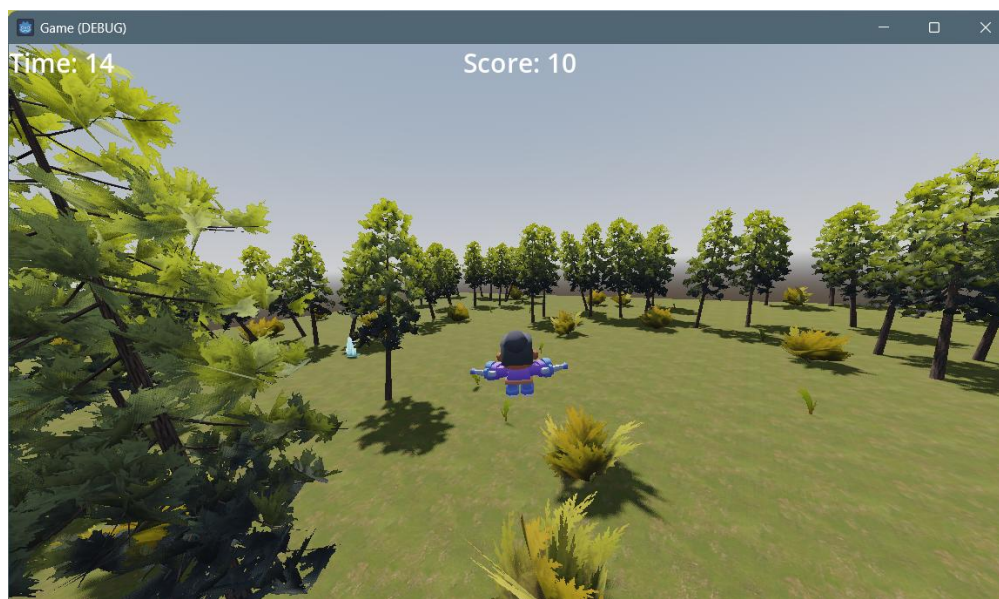
- `Label` за прикажување на резултат (`ScoreLabel`), позициониран на средина на екранот.
- `Label` за прикажување на време (`TimerLabel`), позициониран на лево на екранот.

- Control UI елементи: GameOverUI и WinUI. За прикажување на “Game over” порака при губење на играта или “Победивте!” порака при победа.

Слики од играта:



Почетна состојба на играта



Скокање на објекти



Тајмерот истекува, играта завршува



Играчот паѓа од дозволената површина за движење, играта завршува



Победа на играчот при собран максимален број на богатства

Референци :

- a) <https://www.youtube.com/watch?v=OoVAtGHEgjA&t=957s>
- b) <https://www.youtube.com/watch?v=wO-Va6BpeyQ>
- c) <https://www.youtube.com/watch?v=VeCrE-ge8xM&t=12s>