

COMP 8552

Project Design Document

Team Members and Roles:	3
Engine Design	4
Core ECS	4
Game Design	5
Features (Detailed)	6
Tools	7
Risks & Mitigations	7

Team Members and Roles:

Name (Student #)	GitHub	Responsibilities	Documentation
Jovan Sandhu (A01201367)	Jovan9876	Gravity Screen Wrapping	Design
Richard Mac (A00990553)	richardmac604	Player	Design
Connie Wu (A00993734)	s-Kone	Platforms Springs	Design
Trevor Hong (A01232148)	TrevorHong	Camera Platform Generation	Design
Jerry Han (A01308697)	TheRedLegionRises	Audio UI Environment	Design

Repo: <https://github.com/Jovan9876/GameEngine>

Planning: Discord daily meetings

Engine Design

- Base: Course Material (ECS), with SDL for window/input
- Target: 2D
- Audio: SDL3_mixer
- Platform: Windows

Core ECS

New Components:

- Gravity { float accel; float maxAccel; float currentVelY }
- ScreenWrap {bool wrapX; float marginX }
- ScoreTracker { float maxHeight; int Score }
- Platform {enum Type { Static, Moving, Breakable }; Type type }
- MovingPlatform { Vector2D startPoint; Vector2D endPoint; float speed; bool moveToB }
- BreakablePlatform { float breakDelay; float destroyDelay; bool triggered; bool broken; float timer }
- ScoreTracker {float maxHeight, int score}

New Systems:

- GravitySystem (apply gravity to velocity)
- ScreenWrappingSystem (Wrap entity position from left edge to right edge...)
- JumpSystem (set vertical velocity)
- OneWayPlatformCollisionSystem (allow landing on platform from above; ignore from below; trigger breakable platforms)
- PlatformSpawnSystem (procedural spawn above camera; despawn below)
- CameraFollowUpSystem (camera y only increases)
- ScoringSystem (track max height, update score)
- AudioSystem (play SFX; jump/land/break/spring)

Reused Systems: Movement, Collision (AABB), Render, Animation, KeyboardInput, Destruction

Game Design

Working Title

Alien Ascension (Doodle Jump clone)

Narrative / Theme

You're a stranded alien trying to reach your spaceship by any means necessary, bouncing from platform to platform.

High-Level Features:

- Endless vertical climb, score = max height reached
- Screen wrapping (Left <-> Right)
- Platform Types: static, moving, breakable
- Up-only camera: falling below the camera = game over
- Sound: Background music
- Pausing the game

Platforms & Controls:

- PC (Keyboard)
 - A: move left
 - D: move right
 - Jump: continuously jumping, no input key
 - P: pause

Art & Audio Assets:

- Art from itch.io or wherever found
- SFX (jump, land, break, spring) from itch.io or AI-generated

Features (Detailed)

Player Motion & Feel:

- Horizontal acceleration
- Clamped max speed
- Vertical velocity impulse based on platforms, gravity over time

Platforms:

- Static: default platform (unbreakable, unmovable) just launches the player up
- Moving: default platform with motion either left or right, or up or down, dependent on height
- Breakable: collapses after first land

Procedural Platform Generation:

- Despawn platforms below the screen height
- Platform types are placed depending on the current height of the player (higher up, more difficult platforms)

Scoring & UI:

- On-screen score shown during gameplay

Audio:

- BGM

Tools

- Code: C++, SDL, CMake
- Collaboration: GitHub, Discord
- Art/Audio: itch.io

Risks & Mitigations

Procedural Platform Generation Balancing:

- Start with small gaps in platform placement
- Visual platform spawn debugging

Collaboration / Merge conflicts:

- Small PRs
- Only work on designated systems