COMP 4300 Project Proposal

Game Name: Planet Hopper

Genre: 2D Platformer

Overall Theme: Sci-Fi/Space

Main Gameplay Style: Player has to navigate linear levels using platforming techniques while fighting different alien enemies

Overworld: Player controls a spaceship and they can fly to different planets on the screen which represent the different levels

Collisions:

- Player sprite and platforms/tiles that make up the level
- Bullet collisions between player bullets and enemy
- Bullet collisions between enemy bullets and player
- Bullet collisions between tiles/platforms, some tiles can be broken
- Player can collide with collectable items and pick up those items
- Player can collide with hazardous tiles that harm the player

Bullets / Weapons:

- Weapons: ray gun / gravity bomb / space blaster
- Ray gun bullets: A colorful projectile
- Gravity bomb: A throwable projectile
- Space blaster: A missile shaped projectile

NPCs

- NPCs will be enemies that are different alien sprites
- The aliens will shoot at the player / patrol different areas of the map / battle with the player

Moving Tiles

- Different platforms/tiles will move around the map in specific patterns (Might add elevators in certain areas of the map)

HP / Damage

- Enemies will have health bar that appears above them only when they take damage, and decreases with the more damage they take
- The player will have a health bar located on the screen that can be viewed at all times

Status Effects

- Player will always be able to flip their gravity but there will be a status effect to remove that ability temporarily in certain situations
- Collectable shield that lasts for a limited amount of time
- Collectable item to increase player damage for a limited amount of time

Objects / Inventory

- Inventory can be opened and will display player's items they have picked up
- Health packs, Damage consumable, Shield consumable, Movement speed consumable

Ray Casting

- Certain enemies won't react to player until they can see player using ray casting

Lighting Effects

- Dark level with some kind of spherical light that encompasses the player while they progress through that level
- Might use ray casting for lighting effects

Gravity / Acceleration

- Gravity will accelerate player in the direction of their choice (player can flip gravity to accelerate them down or accelerate them up)

Camera / World View

- Camera view centered on player following player through the level
- Minimap of the level

Pathfinding / Steering

- A* algorithm for pathfinding
- Smooth movement using steering (smoothly turn in new directions without snapping)

Game Progression

- Second level unlocks after completing first level and so on

Save/Load Game

- Game data will be written to a text file upon saving in a way that can be read back when loading so that the game continues where it was left off

Shaders

- Whenever player/enemy is hit with a bullet the sprite will be tinted red temporarily to indicate they have been hit

Parallax

- Different levels will have different background layers and the background layers will scroll across the screen at a slower rate than the foreground to create the illusion of depth

User Interface/ HUD

- Player health bar / NPC health bars
- Timer for status effects
- Game progression bar (How close the player is to the end of the level)
- Option menu that can be opened
- Selectable weapons

Sounds

- Different sounds for different weapons
- Sound cue when taking damage / sound cue when hitting an enemy
- Sounds for status effects
- Sounds for killing enemies
- Background music
- Picking up items
- Movement sound effects
- Sounds for clicking on menu buttons
- Sounds for destroying/hitting tiles
- Sound for completing level

Options

- Adjustable volumes for music/sound effects
- Game difficulty
  - Easy (deal 2x damage, 2x max health)
  - Normal (deal 1x damage, 1x max health)
  - Hard (deal 0.5x damage, 0.5x max health)

- Rebind movement keys/inventory key

Extras

- Ability to flip gravity directions
- Sliding on walls / wall jump
- Chain reaction when breaking blocks (Certain blocks cause other blocks to break when they are broken)