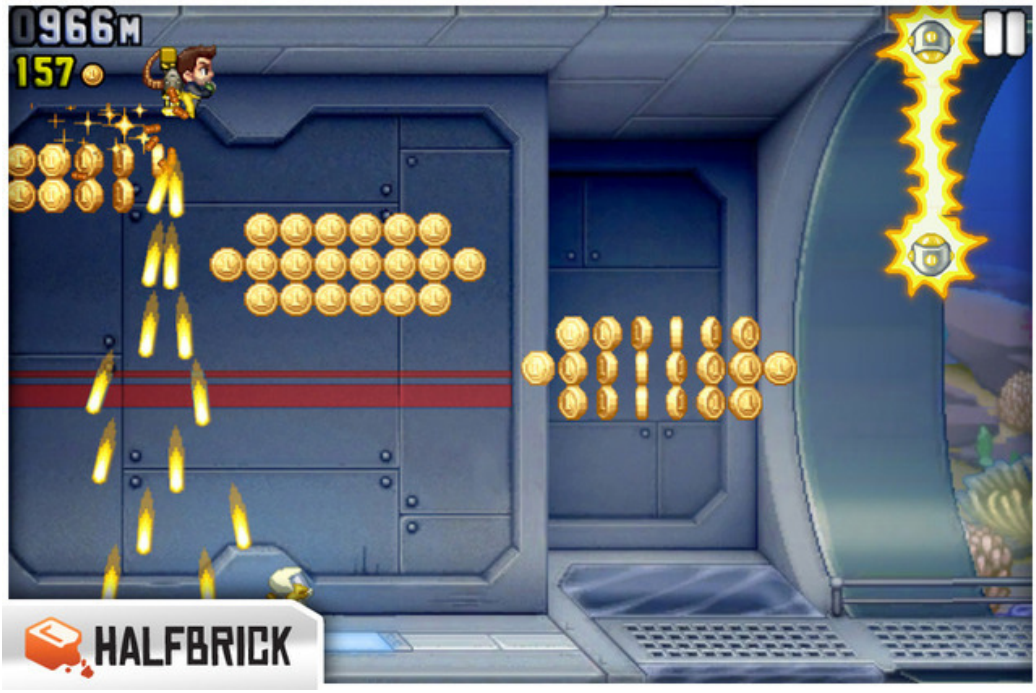
Assignment 1

Computer Graphics

1 General guidelines

Jetpack Joyride is a popular mobile game where the player controls a character using a jetpack to manoeuvre around traps in a lab while collecting coins and aiming for a high score.



This assignment will have you make a clone of this game in OpenGL.

- You are only allowed to use C++ and OpenGL.
- Everything on screen must be modelled with triangles. You may add textures to your game.
- You may use the boilerplate we have provided or start on your own.
- Include a README file with instructions to compile and play your game.

    - Submit your assignment by deleting your build directory, placing all other files in a single .zip file and submitting it as $<RollNumber>.zip$

- Do NOT wait until 2 days before the submission date to ask for help with libraries or any other technical issues. Clear all your doubts as soon as possible. Zero leniency will be granted for extension requests because of this.

- Ask questions about specifications on Moodle, your doubt may help your batchmates as well.

2 General structure

The game should have the player character on the left side of the screen. Obstacles and coins move towards the player creating the impression that the player is moving forward through the level.
The player character can only move up and down to either avoid or collect game objects. (For reference, try to replicate the actual game).

2.1 Levels (15 marks)

- Your game must have three levels, each of a fixed length. Once the player finishes a level, they must be put into the next level automatically. Start at level one.

- Levels must scale in difficulty. How you scale difficulty is up to you.

- Levels must have some visual cues that reinforce the illusion of moving forwards through the level. For instance, you could have alarms on the ceiling, decorations on the walls or tiled floors that move along with traps and coins.

- Reaching the end of the last level should end the game and display a "You win!" message

- Touching an obstacle displays a "You lose" message and ends the game.

2.2 Movement (15 marks)

- Your character should move up and down on the left side of the screen.

- Have your character collide with the floor and ceiling.

- Pressing the space bar (or any other button) activates your jetpack, propelling your character upwards. Letting go of the space bar turns the jetpack off and your character falls to the ground. Your character runs on the floor if no command is given.
- Simulate your characters acceleration due to gravity and the thrust of their jetpack. (i.e. Don't just update your characters y coordinate, model their velocity and acceleration as well!)

3 Obstacles and coins

3.1 Zappers (20 marks)

- The main obstacles players face are zappers.

- Zappers are "lines" of electricity that kill your player on contact. They may be horizontal, vertical or diagonal.
- Spawn zappers in randomly anywhere from the floor to the ceiling. Tune how spawn them in to make your game complete-able. This may be how you scale difficulty.
- Make a small fraction of zappers move instead of being static. You can decide whether they rotate/oscillate vertically/follow the player etc.

3.2 Coins (15 marks)

- Coins are what determines your score at the end of the game.
- Spawn them in randomly.
- Collect coins on contact, removing them from the screen and increasing your score by some amount.

4 HUD (15 marks)

Track the following via onscreen text:

- Level number
- Distance travelled and the levels length
- Number of coins collected/score

When the player wins or loses the game, have a "game-end" screen telling them if they won or lost along with their score.

5 Special effects (20 marks)

To make your game more visually appealing, make the following graphical changes.

- Make zappers glow yellow and you character glow when their jetpack is active.
- (Hint: Use a function such as GLSL's smoothstep to vary how much each pixels color is modified based on how close it is to a glowing object)