Geometry Dash

Charles Chen (cc4919), Riju Dey (rd3054),
Sasha Isler (si2423), Rachinta Marpaung (rbm2168)
Spring 2025

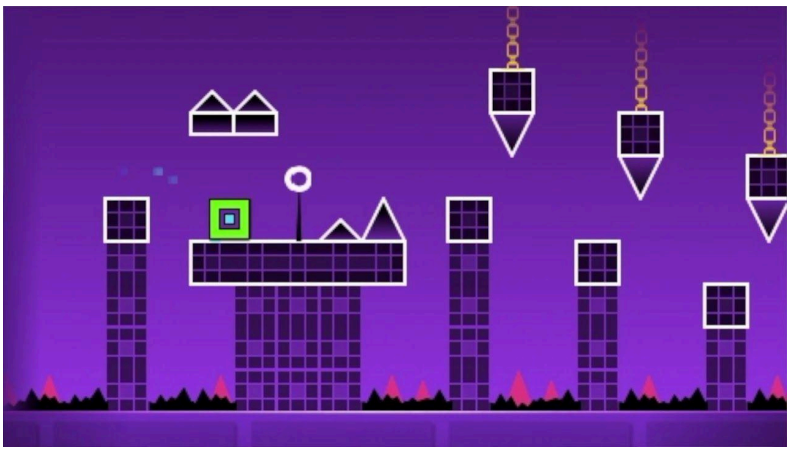

Figure 1: Geometry Dash game

This project focuses on recreating a simplified version of the popular game Geometry Dash. We will leverage the FPGA to handle the game's graphics and user input using the button. The display output will be facilitated through a VGA monitor.

The core of this project is implementing a sidescroller that will be able to handle the game's quick pace. The game will feature vibrant and engaging levels with a multitude of colors. Multiple modes, such as jumping or flying through a level will be supported. An incredible song of choice will be played in sync with the level's leaps and bounds, adding an element of amusement never seen before.

System Design

Framerate will be 60 frames per second. The display will be flicker-free by synchronizing the delivery of new lists of line segments from the software with the screen refresh.

Resolution will be a vga standard with a minimum of 640×480. The final choice of resolution will be dependent on the memory limitations we encounter.

Frame Buffer will be one bit-per-pixel. Frame buffer memory will be on the FPGA.

Hardware-Software Interface will accommodate two main functions: rendering graphics as the sprite advances through the level. This will be done by maintaining the level in an in-memory buffer, a portion of which will be displayed using a conventional side scrolling algorithm. The window that is rendered will be updated at a fixed rate every second. An

additional rendering component will be required to update the position of the user sprite as it responds to input.

User Interface

Input: button on FPGA
Video output: VGA
Audio output: speaker

Major Tasks

- High level decisions about level design and music
- Game logic (sprite movement, collision detection, etc.)
- Implementing side scrolling to render background
- Integrating speakers to play audio of our choice.
- Memory storage techniques, for the background graphics and the audio file
- Implementing speaker