

G54GAM CW2 Documentation

Prototype Design and Specification

Game Overview

RUNxBOW is a 3D first-person platformer, loosely based around the idea of a biathlon. In a traditional biathlon athletes cross-country ski around a track, and after a lap come to a shooting range where they have 5 shots to hit 5 targets, and are penalised for any targets missed. RUNxBOW takes this idea, and sets platforming challenges to traverse and at points in each level has target ranges to attempt as well, with missed targets incurring a time penalty added at the end of the level.

RUNxBOW takes on an 80's neon theme, reminiscent of the aesthetic of films such as *Blade Runner* or *Tron*.

Core game play

The objective of the game is to traverse each level as fast as possible from beginning to the 'end box'. At various points of the level there are 'target ranges' where the player stands on a platform (revealing the targets) and has 5 shots for 5 targets

- When on the platform the player's weapon will appear (a crossbow) along with a crosshair on the UI canvas
- For each platform the player only gets one attempt
 - I.e. if the player comes off the platform the target cover will obscure the targets and not reopen for that level run

If the player completes a level in a new top 3 time and that time will be added to the leaderboard, viewable on the main menu screen.

Game flow

The game is designed to encourage the player to progress at their own speed. Each level is designed to be completed and then reattempted in order for the player to get a better time. The game has been designed to encourage repeated gameplay and

specifically for speedruns to be attempted and continually bettered, with replayability and player enjoyment a main thought behind gameplay

- Progression through the game is only limited by completing a level for the first time
 - I.e. the player must complete level 1 to unlock level 2
- The game is designed to encourage the player to progress at their own speed
- The game features a difficulty curve as levels get progressively harder in terms of obstacles the user faces and penalties for mistakes
 - E.g. In level 2 if the player falls from a platform it will cost them time but the level continues. In level 3 if the player falls from a platform then the level is failed

Characters

The only character in the game is the player. There is no player model as the player is in a first person view for the whole game

- The player can look around using the mouse/trackpad
- The player is able to move using the 'WASD' keys for forward, back, left and right movement, and is able to jump using the SPACE bar
- When on a shooting platform and with ammo left, the player can shoot using the mouse left click
- The player isn't harmed by fall damage/impact with obstacles
 - This allows for the player to attempt large jumps and create falling puzzles [level 3]



Figure 1: Example of falling puzzles

Physics and parameters

The player is bounded by standard physics and gravity, i.e. when falling they will accelerate at the rate of a standard object pulled by gravity over time.

The player will interact with platforms and objects in the level through integrated colliders and so will be 'caught' by platforms below or pushed by obstacles [see level 3]

Level requirements

Levels are used to create separate environments and challenges for the player to overcome. By implementing a level based structure it allows the player to manage progress as well as feed into the 'speedrun' nature as the player can go and replay specific levels to better their time

The game starts with the Main Menu, and the player is initially presented with the only option to play '*Level 0*' (the tutorial level). The tutorial level has floating messages for the player to read as they progress through the level, instructing them on how to perform actions.

Levels increase in difficulty through design but aren't encoded any differently and there is no change in 'base' difficulty through the game.

All features of the game are shown to the player from the outset, the only changes coming in some forms of obstacles such as moving platforms/moving obstacles. Other than this all mechanics are present at the outset

Prototype Instructions

These instructions will show a demo through Level 0 and then Level 1 to demonstrate the main gameplay through a level, the highscore showing on the main screen and then the game proper.

Main Menu

- ☐ Enter unique initials in textbox above Level 0 button
- ☐ Click 'Level 0'
- ☐ Click 'Start'



Figure 2: Main Menu

Level 0

- ☐ Move forward in level to stand on the green platform, revealing the targets
- ☐ Shoot each target in succession using left click
- ☐ Turn left and move forwards, jumping over the gaps in the floor
- ☐ Jump off the end platform into the finish box



Figure 3: Level 0 (Tutorial)

Level End Screen

- ☐ Click 'Main Menu'

Main Menu

- ☐ Click 'Level 0' to show new best time
- ☐ Click 'Level 1'
- ☐ Click 'Start'

Level 1

- ☐ Jump from start platform
- ☐ Stand on green platform
- ☐ Shoot targets
- ☐ Move to left of platform and jump to the lower platform
- ☐ Traverse the floating platforms by jumping between them
- ☐ Shoot the next targets
- ☐ Finish the level

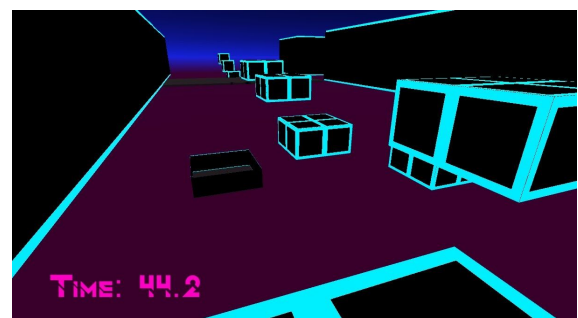


Figure 4: Floating Platforms (Level 1)

Appendix A: External Assets

Within the game some external assets have been used. From the Asset Store some procedural Skybox assets have been taken from the Standard Assets package.

From the Asset Store the crossbow prefab and materials.

From the Asset Store the floating text package as seen in Level 0

From an online source the font “Quantum” used in game

All external assets used are under free usage licenses, and have been separated within the game folders within their own folder ‘Imports’, found within assets