WSOA3004A - Prototype 1

“Untitled Isometric/Infinite Runner Turtle Game”

Joshua Jandrell | Jenna Dunford | Musawenkosi Madlala

PART 1: Task, Scope, Role Assignments, Requirements and Hypothesis

Hypothesis Statement:

“How can we create an isometric infinite runner that uses procedural generation?”

We were given the objective of creating a prototype in groups of three over a week with a hypothesis of our choice. Our group decided to develop a prototype based on the hypothesis statement: how can we create an isometric infinite runner that uses procedural generation?

To begin we laid out the requirements and scope that was expected for this prototype. We would need a character controller, a system that procedurally generated a path for the player and obstacles, as well as a user interface.

Group members roles were as follows:

Jenna Dunford	Creating art assets, building system for generating the isometric path, user interface design, github management, compiling and writing group documentation
Joshua Jandrell	Character controller system design, building system for obstacles and collisions within the game, score tracking, and user event tracking, user interface design, github management
Musawenkosi Madlala	Partial User Interface Design

PART 2: Prototype System, Design Processes, and Methodologies

CHARACTER CONTROLLER

For the character controller, isometric movement had to be considered for the player’s movement. Both forward and side to side movement are along the isometric axes. A constant velocity is applied to the character controller throughout the game in order to make the running “infinite” - the design was so that the player would only need to move to avoid obstacles. A drifting movement was applied to the controller to give a gliding effect to the movement.

For user input - all intuitive inputs were considered and can be used for the player to control the movement of the character.

Similarly to the character controller, the camera also makes use of isometric movement.

COLLISION DETECTION

A collision detection system was used with unity’s tilemap system to detect when the character controller collided with an obstacle. Triggers were used to determine when the character controller should turn.

PROCEDURAL TILE AND OBSTACLE GENERATION

The isometric perspective had to be considered when generating the path for the character controller. Unity's grid and isometric tile map system was utilized. Initially, three lanes were generated procedurally, with varying lengths and a static width - this was implemented along with turning functionality. Parts of the path would generate every set amount of seconds.

The coordinates of the isometric plane were used to generate the path, with initial x and y coordinates being used from each previous part of the path to be used to generate the next part.

Tiles with shark sprites were created to represent obstacles for the player. Four tiles were generated underneath the shark sprite tiles - these tiles contained colliders which would detect when the player collided with a shark.

USER INTERFACE DESIGN

A main menu interface was created to allow the player to navigate between the game, instructions, and credit scenes within the unity build.
Colours were chosen to represent the colour of water - in relation to the ocean and turtle themes of the game (mainly blues).

On the game screen - the user interface involved a button to allow the player to return to the main menu, as well as text that tracked the current and high scores of the player. When the game's lose condition was met - a panel shows up with text explaining why the player lost, and an option to restart the game.

PART 3: Collaborative Development Through GitHub and Online Communication

Methods of communication that were utilized were discord and whatsapp. Discord was used primarily to show progress and share contributions. Whatsapp was used primarily for discussions around commits and conflicts that occured over GitHub.

A GitHub repository was used for collaborative development between group members. Problems were encountered during the project in regards to progress and scripts being lost or corrupted. These problems were resolved.

PART 4: Final Product and Reflections

The final product successfully fulfilled the hypothesis statement. The prototype involved an isometric path that was procedurally generated, and a character controller that could navigate the isometric planes to avoid obstacles. For future development - smoother character controls should be added, as well as save and load systems to keep track of multiple users scores. The path generation script could also be optimized in several places. The turning process at corners could should also be made smoother.
Issues were found in collaborative development with GitHub - in future development, stronger group communication is needed in terms of keeping certain scenes and scripts safe from tampering within Unity - as this can be very destructive to any progress within the project and can undo hours of work.

END

APPENDIX

1. Github repository link:
   https://github.com/jennadunford/WSOA3004-Assignment-1.git

2. References:

Font used: Mouse Memoirs
Designed by: Astigmatic
Found at:
https://fonts.google.com/betterspecimen/Mouse+Memoirs?query=mouse+memoirs
License: Open Font License

3. Art Credits

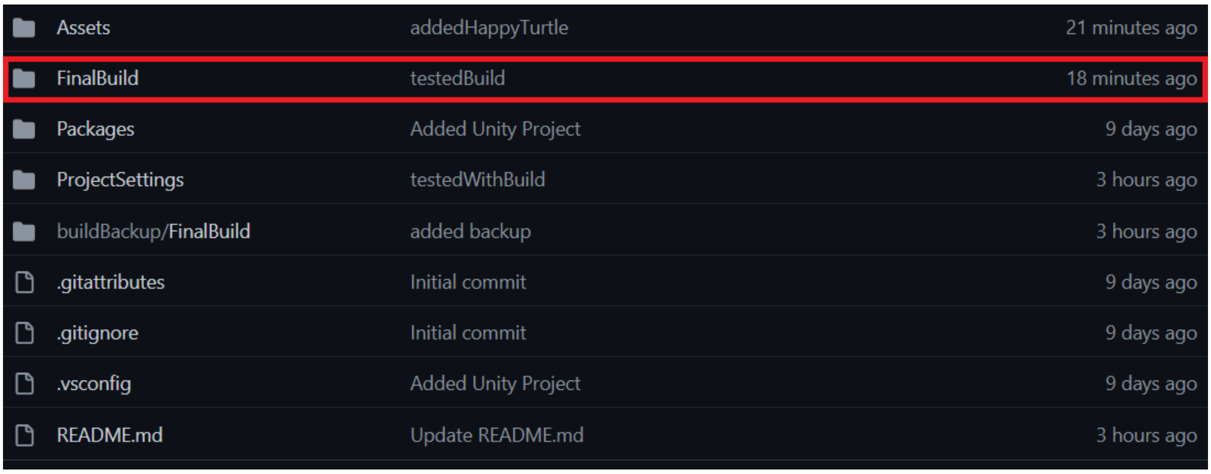
All art assets used within game drawn by Jenna Dunford (2127324)

4. Unity Version:

2021.1.12f1

5. Final Build Location

The final build is located in the Final Build folder within the root of the repository. The image below shows the repository file structure, the folder with the final build is outlined in red.



Assets	addedHappyTurtle	21 minutes ago
FinalBuild	testedBuild	18 minutes ago
Packages	Added Unity Project	9 days ago
ProjectSettings	testedWithBuild	3 hours ago
buildBackup/FinalBuild	added backup	3 hours ago
.gitattributes	Initial commit	9 days ago
.gitignore	Initial commit	9 days ago
.vsconfig	Added Unity Project	9 days ago
README.md	Update README.md	3 hours ago