SE306 Project Plan – Team 'I <3 ROS'

Game Overview

An RPG puzzle game which involves picking up blobs to increase in size and power and attain new skills to help progress through levels.

Target audience:

- high school/university students
- 'achiever' player type - involves levels/clear objectives/high scores

Sub-genre:

- puzzle – use problem solving to complete levels

Overall goal:

- escape the scientist's lab (in which you were created) to freedom, using your constantly evolving set of abilities
- finish levels as fast as possible to gain a higher score

Central character:

- 'Flubber' - a blob of goo which is able to: move up/down/left/right, jump, pick up blobs to increase in size, slingshot movement (i.e. drag back and release to shoot into the air), shoot blobs at enemies, wall jump etc.

Game world:

- start in scientist's lab, progressively increase in size (environment gets bigger too), enemies try to suck up your blobs (e.g. using vacuum cleaners)
- environment includes walls, spikes and other interactive objects (possibly randomly-generated)

Mechanics/dynamics:

- walking/jumping, gradual growth, wall jumping, slingshot movement
- increase in size on picking up blobs, destroy enemies by shooting blobs at them, rebound off walls when slingshot into them
- character blob size will act as the 'life system' i.e. larger model = more health

Design Features

- **High score screen** (5%):
  - score determined by level completion time
- **Monetisation** (5%):
  - freemium model – pay for different character model skins
- **Adding sound** (10%):
  - find free assets which suit the game theme
- **2.5D** (10%):
  - multi-layered parallax background, which scroll at different rates to create illusion of depth

Advanced Features

1. Online leaderboard (10%): We will create a database in the cloud containing the game's leaderboard data for each level. Every time a game is played in which the score is high enough, it will be written to database. The leaderboard will also be viewable through menu selection in which again the database will be queried and loaded onto screen. This will add a competitive element to our game as users can compete with their friends and others online, which caters to our 'Achiever' player type target audience.

2. Replay feature (10%): We will create a database to store player attempts at levels, so they can watch them at a later date, and possibly view replays of top players online (possible integration with online leaderboard – view the top players and watch their replay to see how they did it). This adds an element of competition (either with oneself or other players online) by providing motivation to constantly improve, and increases the 'replayability' of the game.

Tools/Technology

- **Unity:** game engine we will use to create our web-based game
- **MonoDevelop:** IDE we will use for development
- **GitHub Repo:** https://github.com/ElliotWhiley/306_project2
- **GitHub Wiki:** https://github.com/ElliotWhiley/306_project2/wiki - store team information/meeting minutes/documentation etc.
- **GitHub Issues:** https://github.com/ElliotWhiley/306_project2/issues - for keeping track of any problems encountered
- **Trello:** https://trello.com/306project2 - for keeping track of assigned tasks for each different section of the project and discussing implementation specifics for tasks

Work Breakdown and Distribution Plan

Task	**Member/s**
Game world design (environment/level design/story etc.)	Jenny, Toni
Game mechanics (movement/collisions etc.)	Dhanasit, Harvey, Andy
Art design (assets)	Andrew
Online leaderboard database	Namjun
Testing/playtesting	Ross
Documentation/Wiki	Elliot

The above table outlines members who will maintain an overview of the given section, and be the go-to person if other team members need some help in that area. Everyone will still contribute to programming and testing on top of their assigned role.

Iteration 1 (Week 9)

Implement core project foundations:

- Class structure
- Central character design + models
- Level design + generation
- Story development
- Basic world/character mechanics (movement etc.)
- Basic world physics
- Welcome/exit screen

Iteration 2 (Week 10)

Build on our foundations and add extra design features:

- Advanced world/character mechanics (character abilities, life system etc.)
- Advanced world physics

- Scoring system + high score screen
- Achievement system
- 2.5D parallax background
- NPC characters

Iteration 3 (Week 11)
Finish extra design features and implement advanced feature/s

- Online leaderboard/replay feature
- Monetisation – different character model skins
- Sound

Any extra time after the 3rd iteration will be used to finish any underestimated work and prepare for the final demo.

Risk Assessment and Management Plan

Description	Impact	Probability	Management Strategy
Scope creep	Medium	High	Team coding sessions to make sure everyone is on the same page, and cut off dangling tasks as early as possible
Project deliverables not completed before due date	High	Low	Minimum of two meetings per week to receive progress updates from group members on assigned tasks and assign extra members to difficult/lagging tasks
Bugs in project code	Medium	Medium	Ensure all project sections are regularly and thoroughly tested, pair programming whenever possible
Git push which 'breaks the build'	Medium	Low	Work off own branch for new features, push to 'dev' branch once finished, then weekly merge with 'master' branch after code review
Group member not contributing sufficient work	Medium	Low	Monitor 'Hours Log' on GitHub Wiki and commit history to ensure all members actively contribute
Workload underestimated during planning phase	Medium	High	Stay open to alternative design features in case time is running out. Identify time shortages as early as possible during progress updates in meetings
Different group members having different ideas of how to implement tasks	Medium	Medium	Maintain an outline and progress of all current tasks on Trello. 2 group coding sessions every week to ensure everyone is on the same page
Unable to find consistent sets of assets	Medium	Medium	Use basic textured assets (e.g. coloured blocks to represent terrain/environment) if needed