SE306: Project 2 FINAL Checklists

INSTRUCTIONS

Fill in the two checklists as per instructions below and the optional 'additional comments' section as required.

1. Game Features Checklist

> **Planned:** Indicate with a Y/N whether this feature was part of your initial project plan to completed by the final deadline.
> **Coverage by Prototype:** Indicate as a percentage (0-100% in increments of 5) the extent to which this feature had been completed by the prototype (e.g. 0% for not achieved, 50% for half-way, 100% for fully completed) compared to your initial plan.
> **Coverage by Final:** Indicate as a percentage (0-100% in increments of 5) the extent to which this feature has been completed by the final submission.
> **Rationale:** Briefly describe your rationale behind the indicated percentage achieved value by the final deadline (e.g. how can you say it is 75% achieved?) including any relevant examples/evidence.

2. Project Components Checklist

Fill in the coverage (0-100% in increments of 5) the extent to which this project component has been completed along with the rationale.

3. Any additional features or comments (optional, as required.)

Team Name: Team SkyPeak Game Name: Owl Post

1. GAME FEATURES CHECKLIST

Features	**Planned?** (Y/N)	**Prototype Coverage**	**Final Coverage**	**Rationale**
BASIC FEATURES				
Central character/avatar design and functionality	Y	100%	100%	Owl movement and design was done before the prototype submission
Characters design and functionality (e.g. central character and/or enemies etc.)	Y	75%	100%	The main character is the owl and the enemy is the hunter in the park stage – functionality and design completed. The hunter randomly moves and tries to attack the owl with its axe.
Game world layout (e.g. with obstacles and path options)	Y	33%	100%	3 stages and obstacles for each: Park – hunter and terrain (trees and yellow bushes) Mountain – randomly falling stones and terrain (mountains) City – tornado (bounces the owl away) , bad weather which decreases visibility and the terrain (buildings)
A scoring system (point/time)	Y	100%	100%	Score is calculated using an algorithm based on time – the faster you finish a stage, the higher your score
A player life system where lives can be lost/gained/maxed out (e.g. 3 lives max)	Y	100%	100%	Each player has 3 lives for each stage
An achievement system (e.g. rewards	Y	30%	100%	We have 9 possible achievements

unlocked based on player performance.)				that the player can get (such as Mountain King etc). Details can be found in the Achievements wiki page or the Achievements Info link in the game.
A welcome screen (e.g. select a game, return to welcome screen and ability to start again.)	Y	100%	100%	Our map works as the welcome screen which allows the player to access all stages + training stage + battlefield (multiplayer component) + high score + individual profile + achievement info. During the game esc can be pressed to pause. Pause, congrats and game over screens all link back to the Map through the “Exit” button.
An exit screen (e.g. to congratulate player if finished all designed levels or to alert player to indicate game over.)	Y	100%	100%	Congratulation screen for when the player meets the stage objective (collects x amount of mail without dying). Game over screen for when the player loses all 3 lives.
Pre-designed, different levels of complexity	Y	33%	100%	3 stages (easy -> medium -> hard) Park - > Mountain - > City **To indicate the route that the player should take we added the footprints on the map as advised by the tutor in the demo.**
A clear game objective and level objectives (e.g. to get to the end of the level or get 100 points etc.)	Y	100%	100%	Park stage objective = deliver 3 mail, Mountain stage objective = deliver 5 mail and City stage objective = deliver 10 mail. At the start of each stage, the user is informed of their game objective for the particular stage, using a help bubble.
Some aspects of RNG (e.g. random item generation, enemy attacks, level generation.)	Y	50%	100%	Randomly failling rocks, randomly moving hunter, randomly moving tornado, random placement of scrolls (mail) in each stage.
Playtesting of all features within team.	Y	100%	100%	As a group playtested the game. Plus Unity test tools and unit tests were done for the internal code.
Playtesting of all features by at least one other team.	Y	100%	100%	Playtested by and playtested RGB
DESIGN FEATURES				
Major UI redesign (e.g. customizable theme options) up to 10%				
A high score screen (and a mechanism for storing those high scores) allowing users to enter their name for the high score function. Up to 5%	Y	0%	100%	The high score screen shows the top 5 scores for each stage (including the players name). All of this data is stored on a database online – hence it is an online leaderboard.

Adding sound/audio and triggering on appropriate events. Up to 10%	Y	40%	100%	Sound is added to all the stages (background music). Plus trigger sounds are implemented for when the player picks up a mail, loses a life, wins the stage or loses a stage. In addition, sound for wings, rainy weather, wind, the hunter's axe, and the falling stones are also implemented. Volume can also be adjusted by pressing esc and going on the pause screen. Also, the audio can be muted from the same menu.
Local Multiplayer + Leaderboard. 10%				
Online Multiplayer + Leaderboard. 10%				
Touch/Swipe/Tap functionality for those aiming to deploy to a smartphone: Make use of one sensor [worth 5%] or maximum two sensors [worth 10%]				
Fixed level generation. Up to 5%				
Random level generation. Up to 10%				
Monetisation options	Y	0%	100%	A monetisation strategy is discussed in the wiki (for example, adding banners, ads and commercialising the game).
2.5D version of game. Up to 10%				
3D version of game. Up to 20%	Y	33%	100%	The whole game was made in 3D using 3D assets (all stages including the multiplayer component is implemented in 3D).
Tutorial session to help the player 5%	Y	0%	100%	Tutorial session to help the user learn how to play the game.
ADVANCED FEATURES				
Real time onling multiplyer racing game with chat functionality and the ability to create and join different rooms. 10%	Y	0%	100%	The rooms can hold upto 20 players at one time, also the players can communicate with eachother using the chat functionality. The online multiplayer game is real time, which means that the players in the same room can see/collide eachother in the same scene. The purpose behind this feature is to test the flying ability of the player (which is linked to the initial purpose of our game – flying simulation).
Online Individual Profile 5%	Y	0%	100%	Shows a player their highest score in all the 3 stages and also all of the achievements they have earned. We store this data in an online database.

2. PROJECT COMPONENTS CHECKLIST

For the following component, rate overall coverage as follows: 0-25% poor; 25-50% somewhat adequate; 50-75% mostly good; 75-100% excellent

Project Components	**Overall Coverage (0-100%)**	**Rationale**
CODE QUALITY		
Coding Standards (e.g. naming conventions, formatting)	100%	All code was formated commented and named correctly.
In-code documentation/commenting	100%	
Commits and commit comments	100%	
DESIGN CONSIDERATIONS		
Levels are completable	100%	All stages all completable, however city is the hardest to complete.
UI and scoring system clear/intuitive and uses reasonable art/graphics quality	100%	
Not highly repetitive	100%	Each stage has different graphics and new obstacles.
Efficient resource consumption/performance	100%	
Adapts to different screen sizes (mention which ones.)	100%	Can adapt to different resolutions. However, it is meant to be a web game.
Design fits identified user-group (if used)	100%	Casual gamers was our target group and since our game is fast, interesting and not repetitive, it suits our target user group.
TEAMWORK		
Balanced work break down	100%	The whole team worked together in group coding session where they worked on their individual assigned task, yet helped eachother.
Team cohesion and spirit	100%	Everyone worked well together
PROJECT MANAGEMENT		
Implementing the Rational Unified Process as iterative and incremental planning, work, and delivery	100%	
Risk identification and management	100%	
DOCUMENTAION on Wiki (AKA FINAL REPORT)		
Clear mapping of student names with GitHub ids	100%	
Individual contributions per week (e.g. clear list of things each team member contributed to every iteration/weekly)	100%	
Teamwork and project management Approach (e.g. team meetings, how work was co-ordinated, merge/integration of code, risk management, etc.	100%	
Meeting Minutes	100%	
Asset descriptions (hand-made, modified, reused.)	100%	
Design Decisions (SoftEng and Game Design)	100%	
Team Reflections on project (concept, execution), process (rational unified process) and how it fit the game development process, teamwork (what worked, what didn't, areas of improvement), what would you have done differently, future work ideas.	100%	
MISCELLANEOUS		
Extent of Development/Scripting in Unity (e.g. work done from scratch compared to tweaking or use of pre-built components) where 0% means using all pre-built components and 100% means writing everything from	95%	Main ideas for some scripts were from the Unity tutorial sessions.

scratch (its likely to be somewhere in between!)		
Extent of Graphics, Art, and Audio Development (e.g. developed by team members versus used from online resources) where 0% means using all ready-made media and 100% means developing all original.	75%	Basic prefabs (a tree, a rock) from the Asset Store were used to build the 3D game, however, the game world was designed and built by us. All UI screens were built by us.

3. Any additional features achieved or any comments: