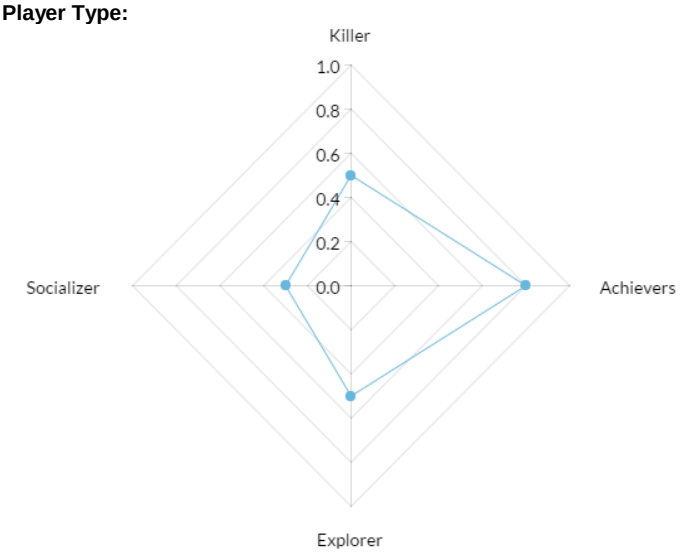
SE306-TeamUGSOFT-Project Plan

Game overview

Target audience: High School(12-18), any gender **Genre**: Stealth **Type**: Simulation
Player Type:



Overall goal: Deliver correct presents without getting caught
Sub-goals:

- Get a good high score
- Collect/eat cookies (bonus points)

Briefly describe your central character (skills, functionality):

- Santa character. Controlled by touch sensitive joystick control
- Skills: Unlock doors
- Unlockable features: e.g. slippers - purchased with in-game points
- Movement (x,y) – no plans to implement vertical movement (i.e. jumping)
- Actions: slow movement, fast movement, crouch, stick to wall

Other important characters:

- Elf: acts like a narrator and tells Santa what to do
- Animals: located around the house may be asleep or awake
- Human: children or adults in the house for Santa delivery

Game world:
Static elements:

- House
- Walls
- Rooms and doors
- Creaky floor boards
- Christmas trees
- Furniture such as tables, beds, chairs, shelves

Dynamic elements:

- Human (awake or sleep walking)
- Animals

Scoring and Lives mechanism:
Scoring is based on three criterions:

- Suspicion level, the lower the better
- Time to deliver the present the shorter the better
- number of cookies obtained

Level generation plan and elements of random generation:

- Fixed rooms
- Start location fixed (chimney base)
- Locked doors
- Random location and movement of humans and animals, although some may have fixed patrolling movement

- Random location for cookie spawn

Level difficulty increases

- more obstacles
- more humans and animals
- more locked doors
- larger house/less hiding places

Design Features

- Fixed level generation 5%
- Adding sound to game 10%
- Implement 2.5D gameplay 10%
- High score screen 5%

Advanced Features

- The advanced feature being implemented is our take on a One Versus One Time Trial match.
- The players will enter a level exactly like how it would appear if they were playing single player. Each player's goal is to complete the level with a higher score than the other.
- Whilst they are playing they will be able to see a projection of their opponent's Santa. The projection will show exactly what their opponent is doing in real-time. This includes seeing which path their opponent is taking, what interactions they are having with the world and even if they get caught and lose the game.
- This feature involves developing code that:
  1. Matches two opponents and then transmits messages between their clients for the duration of a match.
  2. Uses the received messages to render the "ghost" of an opponent.
  3. Identifies the winner based on which client first sent a "finished" message to the server.
- The work required to implement these components easily accounts for the 10% allocated to the Advanced Features.

Tools and Technology

- Windows 7 or higher for operating system of developing environment
- Unity 5.2.0f3 for game development
- Microsoft visual studio for coding IDE
- Google play service for leader board, high score and multiplayer
- Deploying onto android v4.0 or higher

GitHub link (for version control): https://github.com/gdsl/SOFTENG-306-PROJECT-2
Github wiki: https://github.com/gdsl/SOFTENG-306-PROJECT-2/wiki/Documentation
Google doc (for documentation draft):
https://docs.google.com/a/aucklanduni.ac.nz/document/d/1EBKnI7ZLcT8wiNvLAYkhVzqtHpb2DuXj5-KI3KXk3hE/edit?usp=sharing

Work Breakdown and Distribution

Task	Type	Week working on	People
Character design	basic	Week 3	Kevin, Jared and Chang Kon
Other Characters	basic	Week 3 to 4	Jared and Chang Kon
Score and life system	basic	Week 3	Wesley
Achievement	basic	Week 3 to 4	Jay and John
Welcome and exit screen	basic	Week 3	John
Engaging world, aspect of RNG	basic	Week 3 to 5	Guyver, Chuan and Nikhil, Chang Kon
Pre designed level and clear objective	basic	Week 3	Guyver, Chuan and Nikhil
Fixed level generation	design	Week 4 to 5	Guyver, Chuan and Nikhil, Chang Kon
High score screen	design	Week 5	Jay, Kevin

Implement 2.5D	design	Throughout	Everyone
Sound	design	Week 5	Jared, John
Online Multiplayer	advance	Week 5	Guyver, Nikhil, Wesley

Risk assessment and Management plan

Risk	P	I	E	Actions	Warning Signs
Code integration	7	8	56	Integrate often. Plan together and understand other tasks	Merge conflicts.
Burnout	7	7	49	Spread out the development so that more work is done as early as possible	People starting to not turn up to meetings, not contributing as much, and not participating in discussions
Shortfall of externally supplied components	7	5	35	Team must make sure to check that suitable assets with commercial use licenses	Lack of assets belonging to common theme
Complex maintainability	5	7	35	Attempt to keep technical documentation, regarding all modules of the code, up to date for each iteration.	Developers are finding that extending on modules created by other developers is no straightforward task.
Team politics with negative influence on team environment and on the project	4	8	32	Ensure that any criticism or feedback for the teammate or the teammate's work is constructive and not derogative.	People slacking off causing other people to be upset
Boring game	4	8	32	Test often with many different people and get feedback. Use feedback to improve game design.	Play testing different from expectation
Unrealisable Scope definition. Some features may be too difficult with our skills and tools	5	6	30	Plan carefully and do some research and prototyping before putting a lot of resources into key tasks that may seem uncertain	A lot of resources being spent on a task without much progress being made and constantly running into obstacles
Change of scope	4	7	28	Define most important deliverables and concentrate and meeting those features then concentrate on extra features. Have meetings frequently so that members can discuss implementation.	Disagreement in product deliverable features or time constraints restricting initial scope definition
Unfamiliar with tools	9	3	27	Group learning session to learn tools together	People ask question too often about the tools
Variable code architecture	5	5	25	Plan together, discuss domain models together. Form architecture plans early to reduce integration issues later if change occurs.	Change of scope or requirement resulting in different domain model definition 2. Realization that initial domain model not being suitable for implementation
Poor task delegation	4	5	20	Outline tasks and estimated task completion and distribute tasks evenly between members	Members not inputting equal time into project
Getting sick or other health issues	4	3	12	Try to give lighter tasks to the people who are sick so they can rest and recover faster	Sudden cold weather and increase number of flu cases around the university
Computer Crash or broken	2	5	10	Use UG4 Computers while the computer is getting fix or replace. Commit frequently and push.	Depending on what computer problem the sign can be different but more frequent crashing
Family/Other commitment	3	3	9	Try to plan around that and spread task out more equally	The person may say they are away or busy during the weekends
Unable to meet deliverables at due date	1	8	8	Each deliverable can be broken down into smaller tasks and spread appropriately across time.	Sub tasks are not being completed in appropriate time. Doing tasks last minute
Github getting ddos	1	3	3	Make sure work is not left to the last minute	Github being slow and not able to push/pull