Coursework Specification

CSC8508 - Engineering Gaming Solutions within a Team

Final Build - Software Engineering

Due 16:30:00, Friday, 27 Mar 2020

Maximum mark is 25

This is a group exercise.

Game Design Specification: Golf Rules

The challenge is to develop a game across multiple platforms (PC and PS4). The content may vary between the PC and PS4 builds of your game to suit the hardware capabilities, but the central game engine must be shared across both platforms.

The central gameplay mechanic is to use a combination of physics objects and gameplay interactions to create an engaging 3D golf game, where the player must navigate a series of levels by hitting their ball from a starting area to a specified end area. Under normal conditions the player may only hit their ball when it has come to a stop, and the goal is to reach the end area in the least number of hits. The player should be able to set up their next shot using an appropriate on-screen interface that shows the currently selected direction and power.

The game should support some form of multiplayer. This can take the form of split-screen on a single machine (especially PS4), and/or fully online across multiple devices. Unlike a real game of golf, all players play simultaneously in the same game world, such that any side effects of their ball colliding with objects in the world as it moves may help or hinder other players.

Each level should be filled with obstacles that should in some way affect the player's ability to complete it. These can take any form that you wish – for instance they could be seesaws, robots that chase nearby balls, or elevators that must be called to the correct floor by bouncing off a button.

The game specification includes a further twist: There must be multiple ways in which the players can change the fundamental rules of the game as each level is played. For instance, a player could press a button to invert gravity, or collect a powerup that can change everyone else's ball into a cube. A collectable object could even modify the win condition from being who gets their ball to the end area first, to who has hit their ball the most in 10 seconds – the choice is yours, but the more extreme the rule changing, the more interesting your end result will be!

The following videos should help visualise the gameplay elements of the specification:

https://www.youtube.com/watch?v=NyNpPWvw1eU

https://www.youtube.com/watch?v=z3_yA4HTJfs

Features

The game design specification is purposefully lightweight so that, as a team, you can focus on the areas which most interest you – gameplay, lighting, physics, AI, particles, multiplayer, etc.

The design is also purposefully abstract; there is no need to take complex art models from Turbosquid or Blender (no marks are assigned for art content), but does allow for any textures, meshes, or audio that you find to be used as you see fit.

The game must feature:

- A high number of objects making up each level (this is a technical challenge to maintain a good framerate)
- In game HUD to allow each player to see their score, shot power, etc.
- Debug mode must show framerate, timing costs of features in milliseconds, memory footprint, physics collision information.
- The game must pass the Technical Requirements Checklist (see below).

Some optional ideas, but feel free to think of your own (note these ideas are not evenly weighted in terms of marks awarded, as some are much easier to implement than others):

- AI agents that interact with the players.
- Multiple levels.
- Subgames to play between levels.
- Audio effects.
- Multiplayer. At least four online players on PC build (one player per PC)
- At least two players on PS4 build (split-screen on single PS4).
- Android app that allows for modification of a game's rules as it is played.

Further notes:

- The basis for the game code should be the NCL framework you used for previous coursework.
- Middleware can be incorporated into your build, but not engines (i.e. you can't use Unreal Engine or Unity 3D as the basis of the technology). If you choose to use any middleware you need to check the agreement for its use.
- You must use source control (Git - https://git-scm.com/) and continuous integration (Jenkins https://jenkins.io/)
- It is recommended that you also use Slack (https://slack.com/) to aid team communication.
- This is not an exercise in game design, so don't worry too much about making a perfectly balanced game loop. Concentrate on the technical aspects and on bringing all the elements together as a coherent package.
- It is recommended that you take a data-driven approach to development, and that there is a way in which the levels can be quickly prototyped and modified.

Timetable

- An alpha build of the game is due on 6th March.
- The final build of the game is due on 27th March.
- Final build demonstrated and discussed with Gary and Graham in the games lab during the afternoon of Friday 27th March.
- Every Friday, there are a set of milestones to be demonstrated and discussed with Gary in the lab. Note all demonstrations are based on the build on the server – all functionality must be incorporated in the main build. A playable game is required throughout. These milestones are:
    - Week 1 (Friday 31st January 1:00pm)
        - Fast prototypes assessed and game design agreed
        - Roles and responsibilities for each team member defined.
        - Source control and continuous integration in place
        - Middlewares identified.
    - Week 2 (Friday 7th February 1:00pm)
        - Functional prototype of core gameplay in playable form
        - Each code module interface implemented (many with empty or placeholder functionality).
        - Cross platform framework in place, compiled and running on PC and PS4.
    - Week 3 (Friday 14th February 1:00pm)
        - Demonstrate how each code module has been fleshed out, and is utilised cleanly through its interface.
    - Week 4 (Friday 21st February 1:00pm)
        - All middlewares integrated with demonstration of functionality.
    - Week 5 (Friday 28th February 1:00pm)
        - Team defined milestones.
    - Week 6 (Friday 6th March 1:00pm)
        - Alpha deadline – all major functionality in place.
    - Week 7 (Friday 13th March 1:00pm)
        - Team defined milestones.
    - Week 8 (Friday 20th March 1:00pm)
        - Beta deadline – all functionality complete, ready for bug-fixing, TRC testing, and polish
        - Any further features need full risk assessment at this meeting.
    - Week 9 (Friday 26th March 1:00pm)
        - Gold build

- The project should be developed as a team during normal working hours (i.e. more or less 9am to 5pm). Agree core hours (at least 4 continuous hours) when whole team is present every day.
- Hold a brief daily stand-up meeting to agree what the next tasks are, and identify issues. Inform Gary of the time of the daily meeting so that he can attend as an observer.

Team Project Technical Requirements Checklist

In order to pass the Team Project, the following TRCs must be met

- **Framerate**. Framerate on lab PC, in full-screen, must be at least 40fps. Framerate on PS4 must be at least 20fps.
- **No debug info visible**. In default mode, no debugging information should appear on screen – inclusion of a mode which displays debug info is required (e.g. via a key combination), but the default mode should only show what

the player would see. Also no debug info should be printed to the console window.

- **Pause button**. It must be possible to pause, and resume, the game if in single player or split-screen mode. In pause mode any audio must be muted. An on-screen message should show that the game is paused. A “quit game” option should also appear on the pause screen.
- **Loading screens**. The screen should not be blank for more than 0.5 seconds. If required use a loading screen. The screen should never be completely static for more than 0.5 seconds (use an animated icon to show loading is taking place or the game is paused and the game has not crashed).
- **Exit the game cleanly**. Quitting the game, or closing the window, should not cause it to hang, or leave anything running in the background.

Last updated 20 Jan 2020 15:22:02

Submit this exercise