RESILIENCE

2020-2021

Achievement Guide

14th Annual Oregon Game Project Challenge

www.ogpc.info

OREGON GAME
PROJECT CHALLENGE

OVERVIEW

The achievements for this year have been modified to reflect the change to a (probable) remote event. As such, many of the traditional achievements have been boiled down to hopefully streamline the judging process and facilitate remote work by students.

Achievements this year are broken down into multiple categories (similar to the multi-tiered achievements from years past). To qualify for a higher tier, a team must complete all requirements form the previous levels. Even if it would be possible to complete some aspect of a higher level without completing everything from the lower level, a team will only qualify for the lowest completed level.

The achievements are broken into the following levels:

Standard: New teams should try to target the Standard level of every achievement. These are achievements we feel should be easily attainable by students of any skill level.

Encouraged: The Encouraged level is slightly more advanced than the Standard level. These are achievements that most teams should work towards, even if you can't quite get all of them.

Extra Credit: Making a return from previous years, this level is the most advanced. Teams targeting this level will have to really go above and beyond to consistently reach the Extra Credit level.

Programming

Maintenance of Code

Code should be easy to follow, saved in a central location with backups, and have development over time.

Standard

- Code is clean: labelled properly, consistently organized, and named descriptively
- Save multiple copies of the project, or project is saved to the cloud/online

Encouraged

- Code is organized, etc.
- Git/version control history shows at least a few commits over time (save a few versions on a flashdrive, etc)

Extra Credit

- Code is elegant
- General "best practices" were followed and the code would allow for future development
- Git/version control history shows many commits, smaller or even daily commits

Bug Smashing

Show how problems were fixed, avoided, and purged!

Standard

- Talk about or show a logic bug (not typo) and talk about/show how it was fixed
- Game demonstration doesn't crash, freeze, or otherwise completely break

Encouraged

- Talk about a proactive strategy you used to avoid bugs
- Gameplay demonstration is free of major/obvious bugs
- Have a testing process to help search for bugs

Extra Credit

- Have a system to keep track of bugs you've noticed (a list works fine)
- Game executes flawlessly
- Use an automated test system

Dynamic Aspects

Games make use of time or other dynamic elements to enhance gameplay.

Standard

- Game features more than one version of at least one asset to represent different states and dynamically swaps those assets at least once (ex: player can pick up and wear a yellow shirt)
- Animate something by swapping sprites or models

Encouraged

- Game features a time system which meaningfully affects gameplay
- Game swaps/combines/modifies assets or modifies mechanics based on time system (night, seasons, etc. in one level, enemies speeding up as the countdown nears zero)

Extra Credit

- Game dynamically swaps or combines assets to match player actions
- Choose at least one of these three:
    - Game features a full-featured time system integrated into every aspect of the game
    - Game generates content or levels procedurally
    - Game has dynamic or procedural content/level generation

Separation of Powers

Menus allow players to interact with a game before launching into play.

Standard

- Game has at least one menu with an option to view credits
- Game has leaderboard in RAM (clears when game is exited)

Encouraged

- Game has a start menu with options to play, view credits, and change options/settings (at least 2 functional)
- Game has a save system, leaderboard, or other persistent *local* record

Extra Credit

- Game has a pause menu with resume, options/settings, and save

- Game has a save system, leaderboard, or other persistent *online* record (server doesn't need to be hosted)

Bag O' Tools

How did you leverage conditionals/repetition/data structures to implement a mechanic/ diagram logic?

Standard

- Code has loops and conditionals
- Code uses data structures

Encouraged

- Code uses loops and conditionals consistently and properly
- Code uses data structures in multiple locations
- Game logic for at least one mechanic or interaction has been diagramed

Extra Credit

- Game implements an algorithm
- Player state stored in a reasonable data structure (not a group of global variables)
- Use classes or similar data/functionality grouping structures where appropriate
- Game logic for at least two mechanics or interactions has been diagrammed (paper, or using a tool)

Game Design

Design Doc

Write a design document. What are your milestones? What is being communicated to the player? What is the general mood of the game? Etc.

Standard

- Team created a game design document that describes the game at a high level.
- Design document lists milestones ((hopefully) be done by main event!)
- Design doc contains basic info such as win condition, mood, etc.

Encouraged

- Design doc has been updated at least once to show changes in development process
- Design doc has multiple milestones set
- Design doc communicates key game mechanics, the main character, and the setting

Extra Credit

- Design doc has been maintained through development with multiple updates
- Design doc has UI mockups
- Design doc has all important game mechanics, all important characters, the setting, info on all levels

Mechanics

Games task the player with completing various goals however the specific processes and environments in which the player accomplishes those goals can vary greatly.

Standard

- Player is given more than one mechanic for interacting with the game (teleportation, rewinding time, picking up items, jumping, sprinting etc.)
- Game has at least one, well-crafted and well-designed environment (level, biome, map, etc)

Encouraged

- Game has notably different mechanics (jump and double jump are not notably different)

- Game has notably different environments (levels, biomes, maps, etc)

Extra Credit

- Choose at least one:
    - Game contains at least two sets of contrasting mechanics (enabling different play styles)
    - Game has at least one novel mechanic (other than move, jump, pick up, sprint, etc)
- Game contains at least 3 environments (levels, biomes, maps, etc)
- Player can have a markedly different experience on a playthrough

Evolution of Design

How was the game initially conceptualized? What changed when real people playtested? What was axed due to time or other constraints?

Standard

- Show meeting notes indicating changes in mechanics, assets, or other aspects of gameplay
- Game is playtested (play from the beginning to the end)

Encouraged

- Have images showing early game builds with features, assets, characters, etc. that were dropped due to scope or time issues
- Game is changed based on playtesting feedback

Extra Credit

- Have video showing early game builds with mechanics or features that were changed based on feedback from a playtest
- Game has been playtested on at least two separate occasions with changes after each by non-teammates

Target Audience

Who is the target audience/demographic? What concessions had to be made for them? Does the game adapt to the player?

Standard

- Game has at least one target audience (kids like me, adults, my friends)
- Game has concessions or other limitations in place due to the target audience

Encouraged

- Game has at least one specific, and well-defined target audience (middle-aged hard core strategy gamer, teenage casual gamer, elementary age minecraft fanatic, etc...)
- Team held design reviews to ensure game continued to be fun/engaging for the target audience
- Game has been playtested by the target audience(s)

Extra Credit

- Game has been playtested by members of the target audience(s) who are not team members
- Game mechanics, controls, color scheme, etc. all fit the needs of the target audience(s)
- Game has settings (difficulty, color blind mode, etc) to better adapt to different audiences

Feedback

How does the game communicate with the player? How is the player taught new mechanics?

Standard

- Game has at least one indicator for the player's state (health bar, sanity meter, luminance, etc.)
- Game introduces multiple mechanics over time

Encouraged

- Game has at least two indicators for the player's state (health bar, darkness on camera, world assets, etc.)
- Game provides periodic feedback to the player (score, rank, etc.)

Extra Credit

- Game provides specific and continuous feedback to the player in more than two ways (heavy breathing/heartbeat, red or back in corners of screen, health bar, sanity bar, NPC reactions, etc.)
- Game introduces new mechanics slowly and waits for the player to succeed before continuing

Art and Assets

Make Art

Only assets created by team members – not outsourced, purchased, nor commissioned

Standard

- Create 1-3 visual assets

Encouraged

- Create 3+ visual assets

Extra Credit

- All visual assets made by team members

Hey DJ

Only assets created by team members – not outsourced, purchased, nor commissioned

Standard

- Create 1 song (at least 30 seconds)
- Create 1-3 sound effects
- Record at least one line by a team member

Encouraged

- Create multiple songs (at least 30 seconds each)
- Create 3+ sound effects
- Record multiple lines by the team

Extra Credit

- All music created by team
- All sound effects created by team
- All dialog created or recorded by the team

Cohesive Look and Feel

Everything fits together nicely, and nothing stands out.

Standard

- Have at least one color palette (consistent set of colors used everywhere)
- Have at least one soundscape (consistent set of sounds used everywhere)

Encouraged

- Assets are all of a consistent scale, resolution, and style; no assets are unintentionally jarring
- Sounds and music are a consistent volume, quality, and style; no sounds are unintentionally jarring
- Most player actions have sounds and animations associated with them

Extra Credit

- Visual assets have emotional value and inform the player of the world around them
- Audible assets have emotional value and inform the player of the world around them
- All player actions have sounds and animations associated with them

High Fidelity

Game uses particle/atmospheric effects and animated graphics to enhance the game.

Standard

- Game has at least one particle/atmospheric effect (rain, fog, smoke, sparks, etc.)
- Choose one:
  - Game has at least one sprite sheet (4+ frames)
  - Game has at least one rigged and animated model

Encouraged

- Game uses multiple particle/atmospheric effects
- Choose one:
  - Game has two+ sprite sheets (4+ frames)
  - Game has two+ rigged and animated models

Extra Credit

- Game uses a variety of particle/atmospheric effects created by the team to enhance the look and feel of the game
- Game features at least one sprite sheet or one rigged and animated model *created by the team*

Biomes

Players are given multiple environments to explore

Standard

- Create multiple levels/rooms/biomes/etc

Encouraged

- Level/room/biome/etc styles are consistent across asset types (visuals and audio)
- Assets all match the unique style of the level/room/biome/etc (ex: creepy room plays creepy music)

Extra Credit

- Assets allow mixing on the fly for dynamic music or sprites to match player actions or locations (battle music, the horns in Fable, etc)

THEME AND STORY

Use the Theme

OGPC provides a theme, you use it.

Standard

- The season's theme is present in the game

Encouraged

- The season's theme is clearly connected to the story and level design.

Extra Credit

- The season's theme is consistently tied to all aspects of the game – game mechanics, level design, music, etc.

Story Telling

Every good story needs a cast and a good ending.

Standard

- Create a story outline that shows the general shape of the arc
- The story has multiple characters

Encouraged

- The game has a fleshed-out story with multiple sections
- At least one character is developed

Extra Credit

- Story has a clear beginning, middle, climax, and end.
- Story has at least one secondary or supporting character who is developed equally with the main character

Fan Fic

Flesh out the world and story in materials not designed to be integrated into the game itself.

Standard

- Create one additional story in your world that *isn't* present in the game. This story should tie into a character or location that *is* in your game and can be any length
- Create a glossary of characters and locations

Encouraged

- Create one additional story in your world that *isn't* present in the game. This story should tie into a character or location that *is* in your game and must be either 600 words typed or 2 pages of a graphic novel or comic
- Create a glossary of characters and locations *including descriptions*

Extra Credit

- Create one additional story in your world that *isn't* present in the game. This story should tie into a character or location that *is* in your game and must be either 1200 words typed or 4 pages of a graphic novel
- Create a reference guide or manual for important elements of the story, including characters and locations that fleshes out the story, and includes the glossary of all characters/locations.

Scratching the Surface

Games are much more fun when there is a deep story and fleshed out world to explore. Everything for this achievement must be able to be expressed via gameplay as dialog, narration, carvings/graffiti, audio logs, books, etc.

Standard

- Write a background for at least one location or character
- Write at least one world event that happens prior to the game starting

Encouraged

- Write a background for multiple locations or characters
- Write multiple world events that happen prior to the game starting

Extra Credit

- Tie characters, world events, and location histories into at least one playable subplot (even if it doesn't make it into the game)

Storyboarding

Plan out the theme, progression, player choices, and potential endings

Standard

- Create a storyboard for at least one scene in the game

Encouraged

- Create a storyboard for the player's full path through the game (high level for the entire game)

Extra Credit

- Create *detailed* storyboards for all important moments in the game: opening, climax, key turning points, etc.

Management

Team Players

The team got along nicely, communicated well, and overcame challenges together.

Standard

- Team members frequently communicated

Encouraged

- Team members met (virtually: zoom, hangouts, discord, etc.) at least once for a group work session

Extra Credit

- Regular group work sessions
- Team scheduled review sessions and weekly check-ins

Public Relations

Be professional!

Standard

- Team members all wear coordinated clothing (printed, matching colors, or matching styles)
- Team created a development blog with at least one post
- Create business cards

Encouraged

- Development blog contains at least one post from each team member about their respective areas of work
- Create some trinkets (buttons, pins, magnets, clips, etc.)

Extra Credit

- Development blog has multiple entries by all members on the team spanning several months
- Create a physical mascot

Team/Project Management

Use tools to help manage work: issue tracking, milestones, task management and more make software development easier.

Standard

- Team uses some form of task management (Trello, Jira, Asana, Excel, Github Projects)

Encouraged

- Task management uses some form of grouping (tasks and subtasks)
- Tasks are assigned users and state is tracked as work is completed
- Include bug management in with tasks

Extra Credit

- Task management uses some form of hierarchical tasks (ex: Epic -> Feature -> Work Item -> Task)
- Task management is broken into sprints or other milestones
- Add weights/priority to tasks

Checking the Boxes

Complete TMS. All of it. I dare you.

Standard

- Fill out the minimum required fields for an entry (info on team, at least one team member, etc)
- Include a playable link to your game
- Include a basic description of the game
- Leave no remaining placeholder images (team members, team logo, game poster)

Encouraged

- Fill out *all* game entry page details on TMS
- All team members are linked to game entry
- Include more than one sentence for the about section and the instructions section

Extra Credit

- Game entry on TMS has 10+ images (screenshots, concept art, development images), all images have description, team images, full game description (200+ words, think steam store page), instructions, etc.
- All team members have appropriate roles set in TMS and have profile images

Production Value

Required videos exist, are well put together, talk about the game/team/development, aren't too long, etc.

Standard

- Required videos are all submitted on time
- Videos were scripted

Encouraged

- Required videos have well-balanced audio (no one is ear-blastingly loud or unintelligibly quiet)
- Videos include voice over in-game footage
- Videos were edited (no mistakes or blunders)

Extra Credit

- Required videos have *reasonable* special effects (videos, fades, etc.)
- Videos were designed with a storyboard
- Create at least one additional trailer or promotional video (at least 1 minute long)

CHANGES AND AMENDMENTS

The Oregon Game Project Challenge reserves the right to change, amend, modify, suspend, continue, or terminate any or all rules and regulations of the main event either in an individual case or in general, at any time without notice.

Change Log:

November 2020 – First release