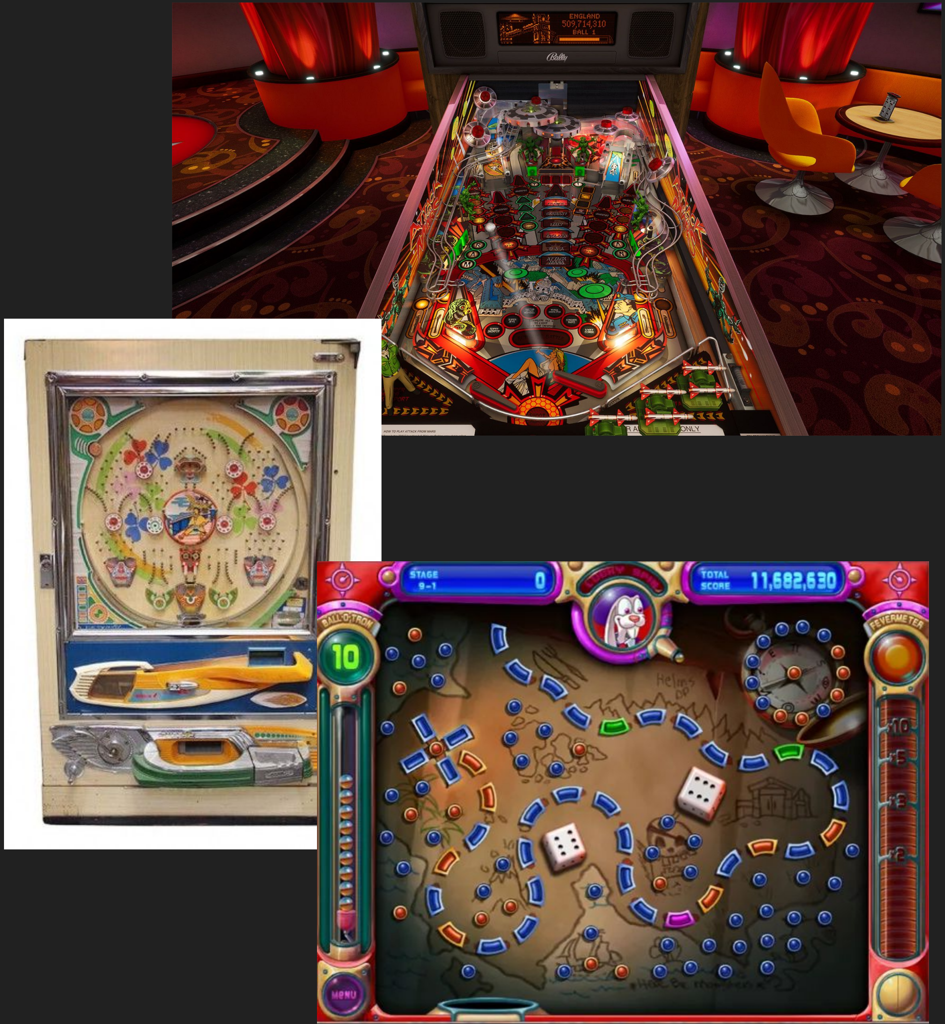
Second Round Presentation

CSC 481 - Computer Science Senior Seminar I

Michael Weiss, Nikolaos Komninos, Tahir Peele

General Project Proposal

Our group is looking to make a spiritual successor to the game "Peggle" by Popcap, a puzzle game inspired by Japanese pachinko machines. The game is simple by nature, with the goal of hitting all orange pegs with the limited number of balls you recieve (think pinball, but you need make sure to hit certain parts of the board). Considering how simple it is, our goal is build the game in the Construct 3 engine in order to port it to web browser, and expand the core game by adding an in-game level editor which allows users to submit levels for infinite replayability.



Gameplay Demo Video

Problem Description

Problem: A new entry in the *Peggle* franchise has not been released in over a decade, and not many other major physics-based puzzle games have been released to fill the niche, leaving the market relatively empty.

Solution: Create a new physics-based puzzle game, inspired by and acting as a spiritual successor to *Peggle*, featuring the same core gameplay that fans know and love, while also incorporating new features to keep players coming back for more.

Initial Research - User Stories

Some possible user stores that research was based on:

As a gamer, I want to be able to play a physics based game that is reminiscent of Peggle anywhere, anytime with endless replayability.

As a casual gamer, I want a game that provides quick sessions so that I can play in short bursts when I have free time.

As a competitive gamer, I want a fun, replayable challenge that will allow me to go back to try and beat my high scores

As a content creator, I want to be part of a community where I can design and share levels for others to play, and receive feedback for them.

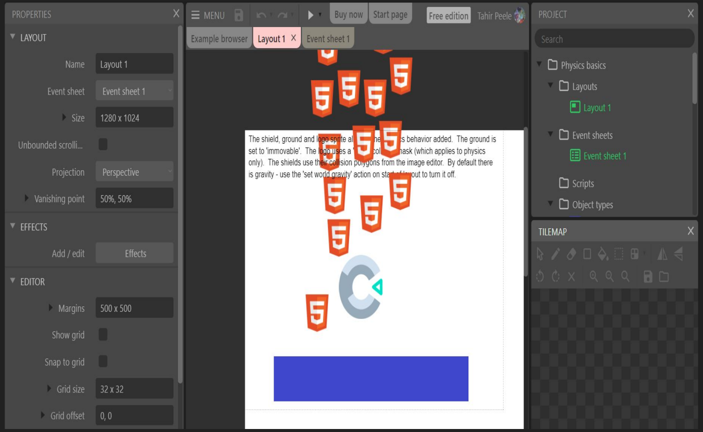
As a community member, I want to relate to others through the levels we all play and create.

High Level Design Concepts - Game Engine

Game Engine of Choice: Construct 3

Why?: Construct 3 includes a robust Box2D-based physics engine, which handles realistic gravity, collisions, and ball bouncing. This is of course important for a game such as this, which places a strong focus on using the physics of the ball to hit as many of the orange pegs as possible using only that singular ball.

The physics system allows for accurate ball movement, making it easier for us to replicate *Peggle*'s trajectory-based gameplay.



LAYOUT
Name Layout 1
Event sheet Event sheet 1
Size 1280 x 1024
Unbounded scrolli... □
Projection Perspective
Vanishing point 50%, 50%

EFFECTS
Add / edit Effects

EDITOR
Margins 500 x 500
Show grid □
Snap to grid □
Grid size 32 x 32
Grid offset 0, 0

The shield, ground and logo sprite al[illegible] he[illegible]s behavior added. The ground is set to 'immovable'. The logo uses a '[illegible] col[illegible] mask (which applies to physics only). The shields use their collision polygons from the image editor. By default there is gravity - use the 'set world gravity' action on sta[illegible]out to turn it off.

- Physics basics
  - Layouts
    - Layout 1
  - Event sheets
    - Event sheet 1
  - Scripts
  - Object types

High Level Design Concepts - Functional Requirements

Gameplay

- The game must have a main menu to choose between playing and editing

- The game must allow the player to launch balls using a launcher

- The game must have orange pegs that need to be hit to complete a level

- The game must have blue pegs that acts as obstacles

- The game must have red (negative) pegs, which destroy the ball on contact

- The game must have a moving basket at the bottom of the level that will provide an extra free ball if shot into

- The game must implement realistic physics-based movement for the balls, including gravity, bouncing and collision detection

Level Editor

- The user must be able to create custom levels using a drag-and-drop interface

- The level editor must allow players to place and remove pegs, obstacles, and other interactive level objects

- The user must be able to upload custom images for level backgrounds

- The user must be able to add custom sound effects to their levels

- The level editor must support exporting and importing levels in JSON format

User Experience

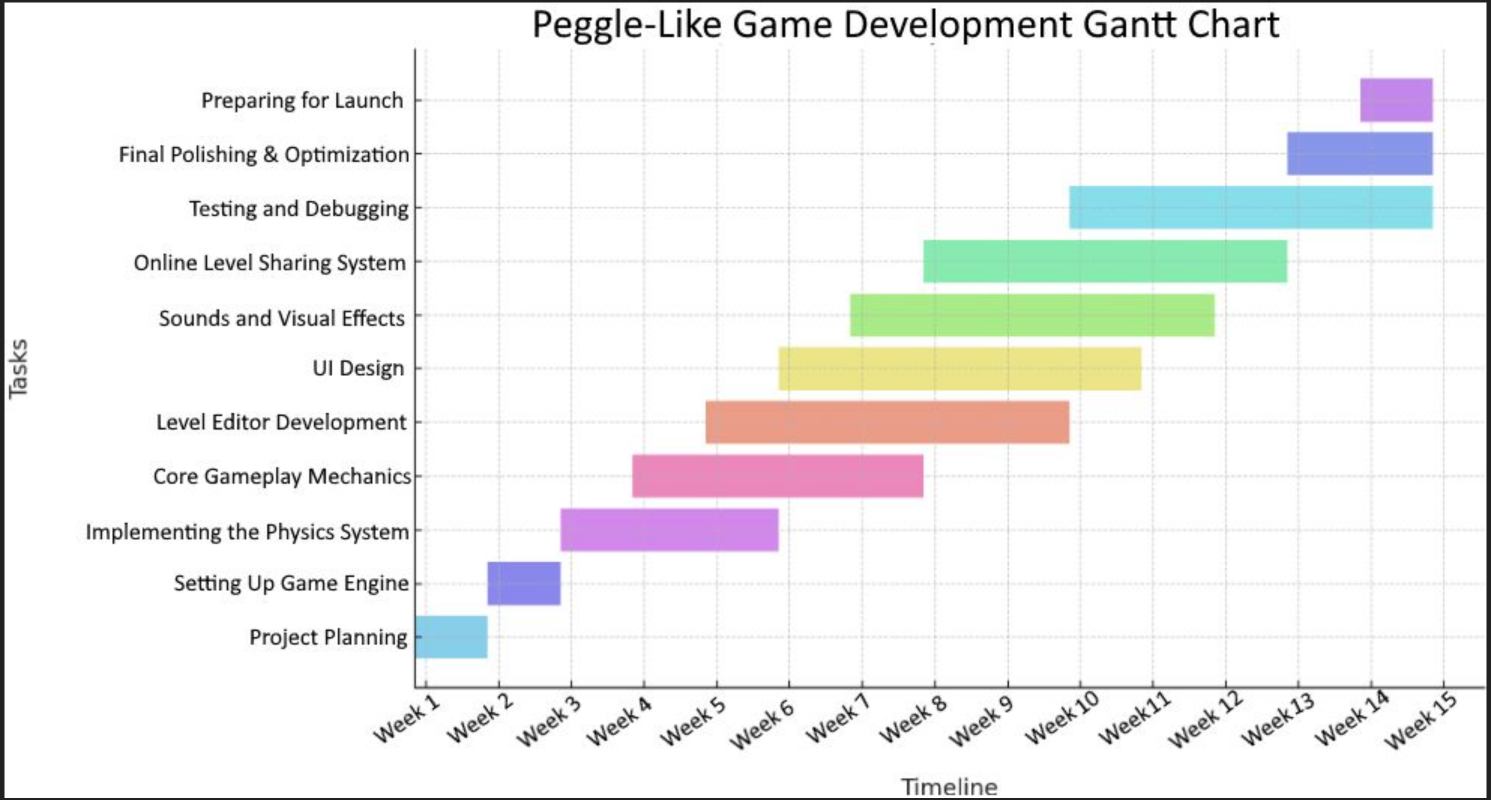
- The game must provide positive reinforcement with sound and visual effects after beating a level (of the celebratory variety)

- The game must reward skill with extra balls or bonus points

High Level Design Concepts - Non-Functional Requirements

Performance	Usability / Accessibility	Scalability
- The game must run smoothly in web browsers without significant lag - The physics engine must remain consistent with the frame rate - The game must load levels quickly / in a timely manner	- The game must support all modern web browsers - The game must be compatible with various screen sizes and resolutions - The game must have a simple / easy to use interface for both gameplay and level editing - The game must have simple controls using the mouse	- The game should be designed in a way that will allow for updates adding new content to be developed with ease - The game should be designed so that updates do not break levels created in previous versions of the game

Tentative Project Schedule



Tentative Project Schedule - MoSCoW

Must-Have

- Accurate physics based gameplay
- Pegs to interact with for score
- Pegs that are obstacles
- Pegs that are needed to win
- Pegs that destroy the ball
- Moving basket to save ball
- Level Editor to create, import and export user-made levels
- Add and delete placed assets in Level Editor
- Externally exporting and importing user-made levels
- Level to play to demonstrate proper gameplay functionality

Should-Have

- Menu to select various levels/layouts for testing
- Drag and drop interface for Level Editor
- A few levels to demonstrate proper gameplay functionality
- Pre-made custom levels made in level editor to show that our game properly loads such

Could-Have

- Placeholder assets (audio/graphics from *Peggle*)
- Ability to import/load custom assets (audio, pictures, etc) into Level Editor
- Saves score for levels and custom-levels
- Unlocking levels via beating the prior level
- Bugs that need to be fixed

Won’t-Have

- Online functionality for importing and exporting user-made levels
- Large supply of pre-built levels to play
- Fully original soundtrack and graphics, and finished presentation
- Peggle*-Masters (characters who alter/change gameplay properties)

THE END.