

Platypus Out of Clay

6610545421 Panthut Ketphan

1. Project Overview

Platypus is a 2D side-scrolling shooter game where the player controls a triangle spaceship, battling waves of enemies, the game features power-ups, challenging enemy patterns. Player will navigate through multiple levels, dodging and eliminating enemies to progress

2. Project Review

A similar existing game is Platypus, a side-scrolling shooter known for its unique claymation art style. Our project builds upon the original by adding:

- More diverse enemy attack patterns.
- An endless mode for replayability.
- Additional power-ups for varied gameplay.
- More detailed scoreboard and performance reports after game over.

3. Programming Development

3.1 Game Concept

Game Mechanics:

- The player controls a triangle spaceship, navigating through levels filled with enemy spaceships.
- Collecting power-ups upgrades the ship's weapons and abilities.
- Defeating waves of enemies and bosses allows progression.

Key Features:

- Diverse enemy AI and attack patterns.
- Endless mode for high-score challenges.
- A variety of power-ups affecting gameplay.

Gameplay:

- The game starts upon launching the program.
- Controls: WASD for movement, Space bar to shoot.
- Enemy types:
 1. **Standard Enemy:** Spawns from the right side and moves left, shooting at the player's current position.
 2. **Tracking Enemy:** Shoots a missile that slowly tracks the player for 3 seconds before exploding.
 3. **Wave Enemy:** Moves in a sine-wave pattern while dropping bombs.
 4. **Boss:** Appears after 2 minutes based on elapsed time. Summons reinforcements on the right side of the screen and can use all enemy weapon types.

Power-ups:

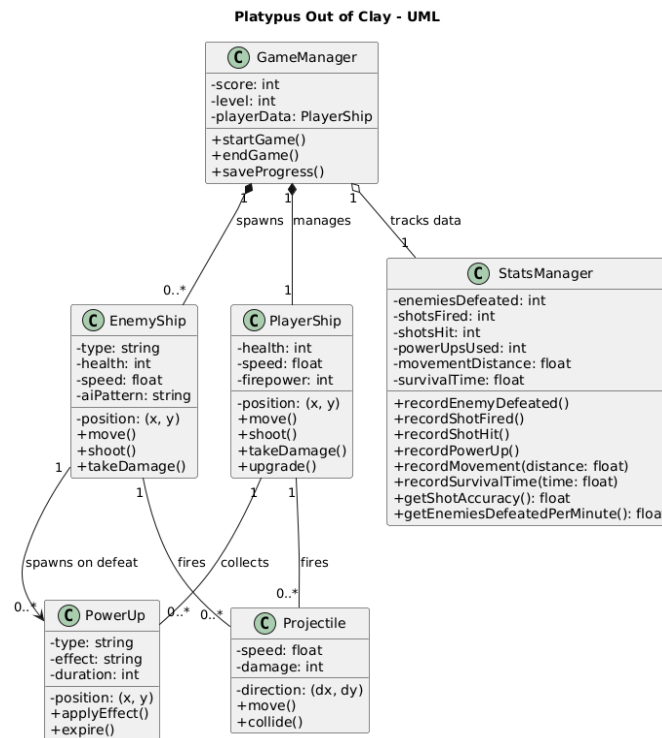
Power-ups are summoned in the form of a star, which players can shoot before collecting to change its type. Each power-up lasts for **10 seconds** before expiring. Additional time can be gained by collecting more stars. The power-up types include:

1. **Shotgun Mode (SM)**: Fires multiple spread shots at once.
 2. **Increased Fire Rate (IF)**: Increases the speed of projectile firing.
 3. **Beam Mode (BM)**: Fires a continuous laser-like beam, similar to the Death Star's super laser.
 4. **Missiles Mode (MM)**: Launches guided missiles that use **Finite State Machines (FSM)** to track and hit enemies.
- Stars have a chance to spawn when an enemy is defeated.

3.2 Object-Oriented Programming Implementation

Classes:

1. Playership
 - Attributes: health, Speed, Firepower, Position
 - Methods: Move(), Shoot(), TakeDamage(), Upgrade()
2. EnemyShip
 - Attributes: Type, Health, Speed, AI pattern, Position
 - Methods: Move(), Attack(), TakeDamage()
3. PowerUP
 - Attributes: Type, Effect, Duration, Position
 - Methods: ApplyEffect(), Expire()
4. GameManager
 - Attributes: Score, Level, PlayerData
 - Method: StartGame(), EndGame(), SaveProgress()
5. Projectile
 - Attributes: Speed, Damage, Direction
 - Methods: Move(), Collide()
6. StatsManager
 - Attributes: enemiesDefeated, shotsFired, shotsHit, powerUpsUsed, movementDistant, survivalTime
 - Methods:
recordEnemyDefeated(), recordShotFired(), recordShotHit(), recordPowerUp(), recordMovement(), recordSurvivalTime(), getShotAccuracy(), getEnemiesDefeatedPerMinute()



3.3 Algorithms Involved

- Sorting: Leaderboards utilize quicksort for ranking scores.
- AI Behavior: Finite State Machines for enemy attack patterns.
- Collision Detection: Bounding box and pixel-perfect detection.
- Event-driven Mechanics: Handling power-ups and player interactions

4. Statistical Data (Prop Stats)

4.1 Data Features

1. Player Movement Data: Tracks total distance traveled
2. Shot Accuracy: Measure hit/miss percent for projectiles fired.
3. Power-up Usage: logs frequency and effectiveness of power-up
4. Enemy defeat Rate - Records number of enemies defeated per unit distance
5. Game Durations: Track playtime per session.

4.2 Data Recording Method

- Data will be stored in a CSV file
- Leaderboard data is updated after each game session.
- Power-up effectiveness is measured by its impact on survival time and score boost.
 - **Survival Time:** Measure how much longer the player survives with the power-up.
 - **Score Boost:** Track score increases while the power-up is active.
 - **Enemy Defeat Rate:** Count enemies defeated per unit of time with the power-up.
 - **Usage Frequency:** Track how often each power-up is used

4.3 Data Analysis Report

- Summary statistics for shot accuracy, survival time, and power-up usage.
- Graphs visualizing player performance trends.
- show how many enemies you can destroy in one sectionsections.

5. Project Timeline

Week	Task
1 (10 March)	Proposal submission / Project initiation
2 (17 March)	Full proposal submission
3 (24 March)	Complete the player side with all type of weapon
4 (31 March)	Complete the star and enemy side
5 (7 April)	Complete visualizations of graph
6 (14 April)	Submission week (Draft)

6. Document version

Version: 4.0

Date: 31 March 2025

7. Data collection

Feature	Why it is good to have this data? What can it be used for	How will you obtain 50 values of this feature data?	Which variable (and which class) will you collect this from?	How will you display this feature data (via summarization statistics or via graph)?
Player Movement Distance	Helps analyze how actively players explore or evade. Useful for difficulty balancing.	Log player position every second and calculate total distance moved per session	<code>position</code> in PlayerShip (tracked in Stats or GameManager)	Graph: Line chart of distance per session
Shot Accuracy	Indicates player's skill level and weapon effectiveness. Can be used to improve power-up tuning.	Track total shots fired vs. shots hit each session. - For more granular data, record shots fired/hit every minute.	- Projectile class (on collision for hits) - PlayerShip for shots fired - Summarized in Stats or GameManager	Summary Stats (mean, median, std. dev) <i>Optionally</i> , a line chart showing accuracy per minute
Power-up Usage Frequency	Shows which power-ups are most preferred or neglected. Helps in balancing game design	Log each time a power-up is collected and its type.	<code>ApplyEffect()</code> in PowerUp class (send data to a Stats manager)	Graph: Bar chart (frequency of each power-up type)

Survival Time	Reflects overall player skill and game difficulty. Good for progression tracking.	Record time from <code>StartGame()</code> to <code>EndGame()</code> for each session.	GameManager (start/end timestamps)	Summary Stats (min, max, average)
Enemies Defeated	Indicates game challenge level and player performance. Used to adjust enemy AI.	- Increment count whenever an enemy's health reaches 0 in <code>TakeDamage()</code> . - For more detail, record <i>Enemies Defeated per minute</i> to normalize for session length.	- <code>TakeDamage()</code> in EnemyShip - Aggregated in Stats or GameManager	Graph: Histogram of enemies killed per session <i>Optionally</i> , line chart of enemies defeated per minute
Score per Session	Core measure of player success. Used for leaderboards or progression.	- Log final score from each session.	<code>score</code> in GameManager	Summary Stats + Graph (leaderboard or line chart over sessions)
Power-up Effectiveness	Helps evaluate which power-up gives the most value (survival time boost, score boost).	- Track survival time or score before/after power-up usage. - Compare differences across multiple uses.	<code>ApplyEffect()</code> and <code>Expire()</code> in PowerUp Linked with GameManager or Stats for survival time & score changes	Graph: Bar chart comparing different power-ups (effect on survival time, score, etc.)
(New) Enemies Defeated Per Minute	Normalizes performance across short vs. long sessions. Helps understand pacing and difficulty over time.	- Each minute, record how many enemies were defeated.	- <code>TakeDamage()</code> in EnemyShip - Summarize per minute in Stats or GameManager	Graph: Line chart (time vs. enemies defeated per minute) or summary stats (avg, peak, etc.)
(New) Shots Fired/Hit/Accuracy Per Minute	Gives a more detailed, time-based breakdown of weapon usage and player skill.	- Tally shots fired and hits each minute.	- Projectile class (on collision for hits) - Summarize in Stats or GameManager	Graph: Line chart (minutes on X-axis, accuracy on Y-axis) or summary stats (per-minute average)

8. Graph Details

Feature Name	Graph Objective	Graph Type	X-axis	Y-axis
Graph 1: Player Movement Distance	Showing total distance traveled per session	Line chart	Session ID (or time)	Distance (units traveled)
Graph 2: Power-up Usage Frequency	Showing usage frequency for each power-up type	Bar chart	Power-up Type	Frequency (times collected)
Graph 3: Enemies Defeated	Counting enemies (or types) killed per session	Histogram	Enemy Type (or Session)	Kill Count
Graph 4: Power-up Effectiveness	Comparing effectiveness (e.g., score boost, survival boost)	Bar chart	Power-up Type	Effectiveness Metric (score/time increase)
Graph 5: Enemies Defeated Per Minute	Showing how many enemies are defeated over time (normalized per min)	Line chart	Time (minutes)	Enemies Defeated per Minute
Graph 6: Accuracy Over Time	Tracking shots fired, hits, and accuracy per minute	Line chart	Time (minutes)	Accuracy (shots hit / shots fired)

Date	Name	Description of Revision, Feedback, Comments
13/3	Rattapoom	<ul style="list-style-type: none"> Needs more explanation on endless mode Pathfinding algorithms How to record “effectiveness” of power-ups Formatting issue: missing section 5 (project timeline) and 6 (document version).
15/3	Phiranath	The Data Recording Method, Data Analysis Report, Project Timeline, and Document version are missing. Don't forget to add the part after 4.1 in the next proposal.

27/3	Rattapoom	<ul style="list-style-type: none">• Please fix your UML diagram to use the standard notations• For data collection, if your game is long, the process of collecting data manually by playing 50 times may be tedious.• You might want to collect something like “The amount of enemies defeated per minute” or “The amount of bullets shotted/hitted/accuracy per minute”.
------	-----------	--