

ESC - Quit

H - Help



TECHNICAL UNIVERSITY
OF CLUJ-NAPOCA

2018
RoCHI

Find the monolith!
But beware of the enemies...

Game Development Methodology Mapped on the EvoGlimpse Video Game Experiment

Bianca-Cerasela-Zelia Blaga

Dorian Gorgan

Health 300

Introduction

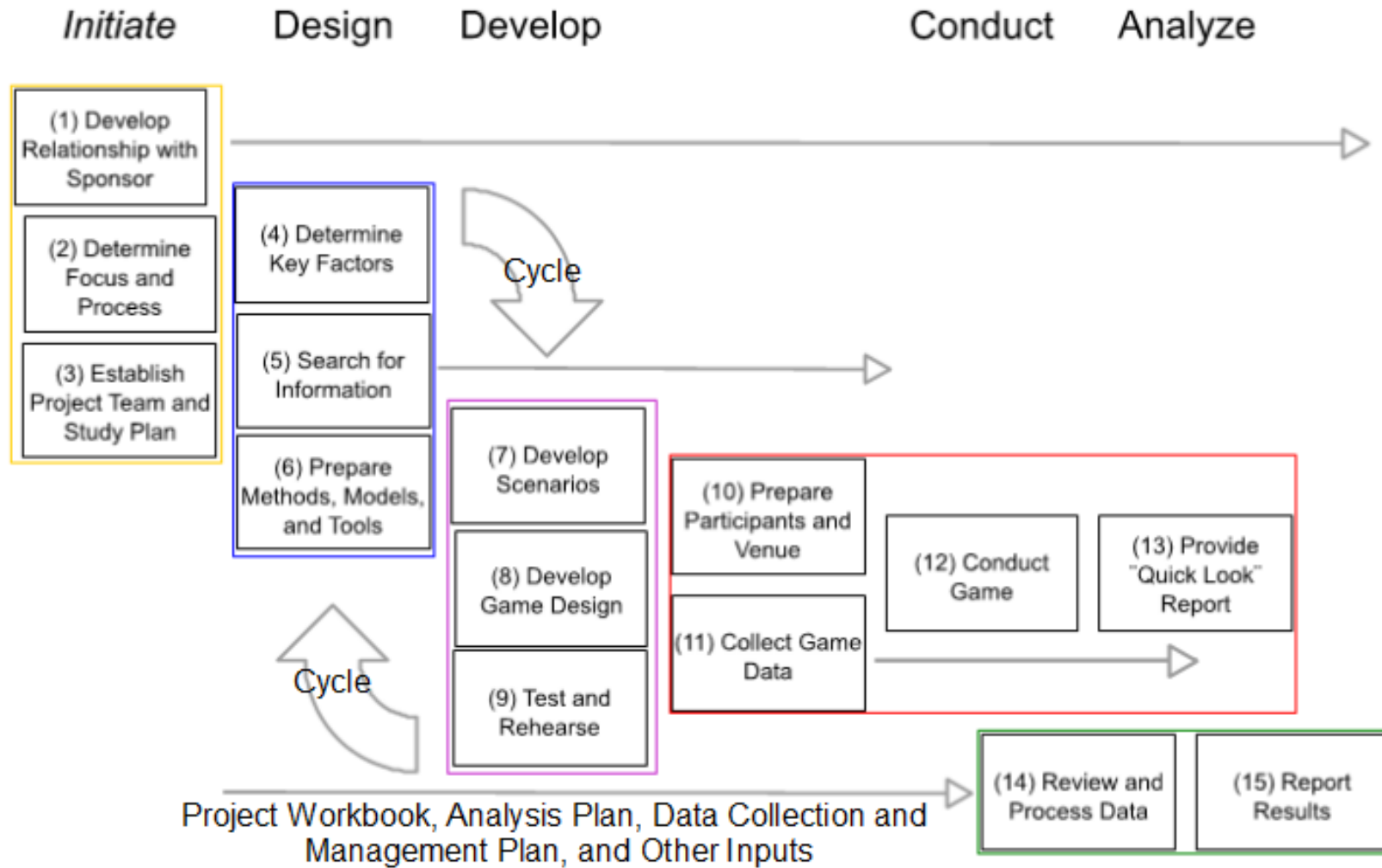
- **Video games** were defined by K. Salen as “a system in which players engage in artificial conflict, defined by rules, that results in a quantifiable outcome” [1]
- **Interactive applications** – interaction between a human and a digital device
- High level of difficulty in developing because of **various elements** such as 3D objects, collision detection, scripting, sound management, animation, rendering, control, and artificial intelligence [2]
- Main objective is obtaining a **high level of usability** [3]



- **Goal:**
 - gain knowledge in game development with a hands-on experiment
 - obtain a highly functional and usable game

Related Work

- [4]



Game Development Methodology

1. Analysis:

- **Topic selection** – in which the inspiration sources are presented;
- **Game specifications** – this includes user requirements and usability specifications;
- **Prototyping** – which helps explore different solutions by creating low-fidelity user interfaces;
- **Scenario and task description** – where the final components of the game are established; steps to establish how the game should be played, how it can be lost or won, interactions of the player with objects from the scene etc.;

2. Implementation:

- **Tools** – the software that was used in building the actual game;
- **Game objects** – the 3D models that compose the game world;
- **Implementation details** – how the scenarios and the actions were brought to life;

3. Evaluation – results of the functional and heuristic evaluation, together with solutions to the problems that were found, and future development possibilities.

Game Development Methodology

1. Analysis:

- **Topic selection** – in which the inspiration sources are presented;
- **Game specifications** – this includes user requirements and usability specifications;
- **Prototyping** – which helps explore different solutions by creating low-fidelity user interfaces;
- **Scenario and task description** – where the final components of the game are established; steps to establish how the game should be played, how it can be lost or won, interactions of the player with objects from the scene etc.;

2. Implementation:

- **Tools** – the software that was used in building the actual game;
- **Game objects** – the 3D models that compose the game world;
- **Implementation details** – how the scenarios and the actions were brought to life;

3. Evaluation – results of the functional and heuristic evaluation, together with solutions to the problems that were found, and future development possibilities.

Next: Case study – EvoGlimpse

Analysis – Topic Selection

Main idea

EvoGlimpse aims to give the players a glimpse into evolution from the perspective of an exterior observer, who can travel at different points in time of Earth's existence. This game is heavily inspired by the movie „2001: A space odyssey” [4], in which a civilization of advanced beings helps humans by presenting them ways that can aid in their survival.

Details

The player would be able to travel in these world in different specific shapes: atoms, energy, swimming, walking, driving the car, flying and exploring the space in spaceships. Different obstacles appear, and the player has to overcome them with the current set of skills. The monolith appears to present the way of going from the past to the future.

Final plot

The goal was to create only a world, a futuristic one, on a planet covered by water, in a developed society, with modern architecture and flying cars. The main enemies will be planes guided by artificial intelligence. The player will have to protect itself from them by shooting, for example with bullets, plasma, or laser.

Inspiration



2001: A space odyssey

stands out for the evocative power it has, even though it relies only on a small set of resources and songs [5]



Blade Runner 2049

for the vision about the future and the fluidity of movement of the flying cars [6]



Dunkirk

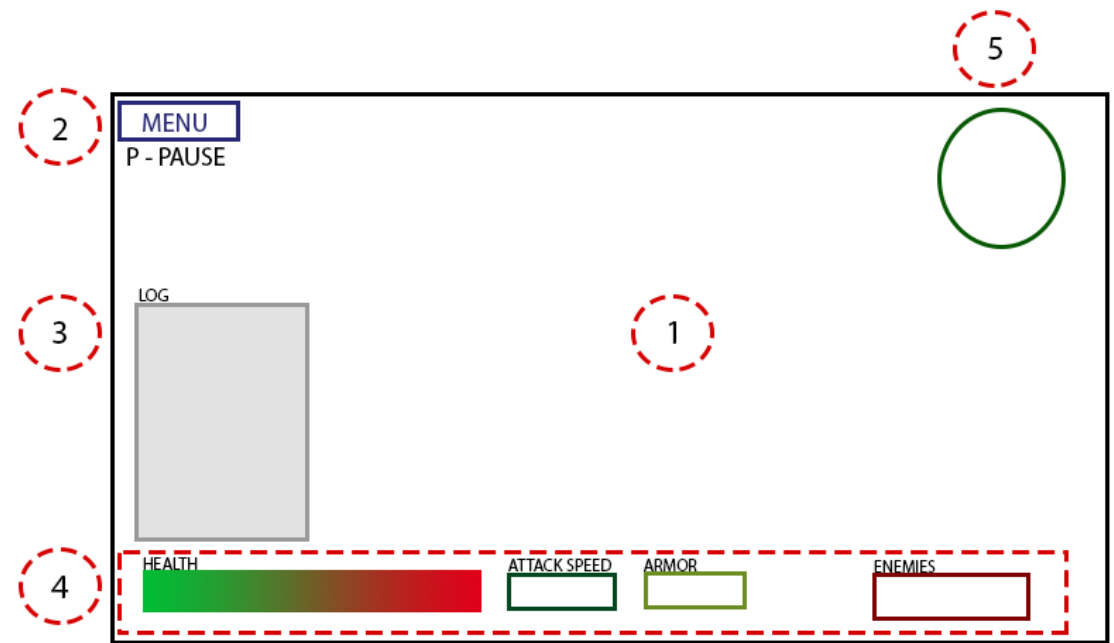
for the scenes in which the planes flew over the water [7]

Analysis – Game Specifications

- Shooter action game for people of ages 12+
- 3 stages:
 1. The player will have some time to get used to the planet and the controls, being able to peacefully explore and observe the world scene
 2. The player will have to protect the planet from some invaders; as the game advances, their abilities increase
 3. The monolith will appear in an unknown location and will have to be found by following its sound signals

Analysis - Prototyping

1. The **main game scene**. Here the movement of the car can be observed. The perspective can be changed.
2. **Menu** and **pause** buttons. The menu contains 5 buttons, each one of them taking the player in a different configuration option.
3. Relevant **logs** for the game. After some options are selected in the menu, a confirmation message will be displayed.
4. Informations about the current state of the game, like **health**, **attack speed** and **armor**. In a window it will be displayed the number of **enemies** remaining and the number of enemies that were taken down.
5. The **map** of the world and the position of the player in it.



Analysis - Prototyping

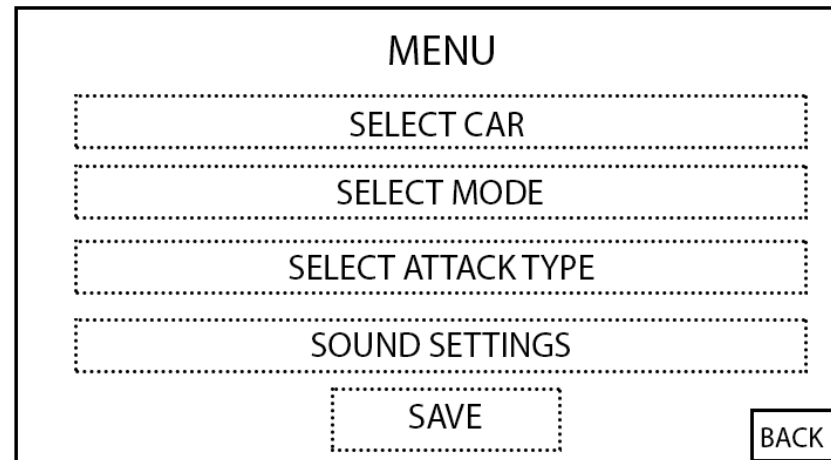
Main game scene that contains:

- Player's vehicle
- Buildings
- Enemies



Menu that enables changing the settings:

- Car selection
- Mode selection
- Attack type selection
- Sound settings



Car selection:

- Green glow – current one
- Yellow glow – when hovering



Analysis – Scenario and Task Description

Scenario	Actions
1. Navigation in the 3D scene	<ol style="list-style-type: none">controlling the vehicle using the mouse movementsincrease speed by pressing spacezoom in and out using the scroll wheel
2. Attacking and avoiding enemies	<ol style="list-style-type: none">observing the enemiesflying towards enemyplayer attacks by pressing the left button of the mousethe enemies attack when the player gets in a certain range and in a certain field of viewobserving the enemies reactionavoiding enemies
3. Monolith	<ol style="list-style-type: none">the player should understand the objective, by reading the message shown on the screensuccessfully navigating in the sceneobserve the monolithfly towards objectivemessage of winning the game
4. Repair power-up box	<ol style="list-style-type: none">recognizing the objectflight towards the objectivecolission with the objectobject destroyedlife health increased
5. Immunity power-up box	<ol style="list-style-type: none">recognizing the objectflight towards the objectivecolission with the objectobject destroyedenemy attack canceled for 20 seconds
6. Display relevant messages	<ol style="list-style-type: none">message with the game objectivestoggle help optionquit buttonplayer health informationmessage of collecting repair power-up boxmessage of collecting immunity power-up boxmessage of destroying enemymessage of losing the gamemessage of winning the game

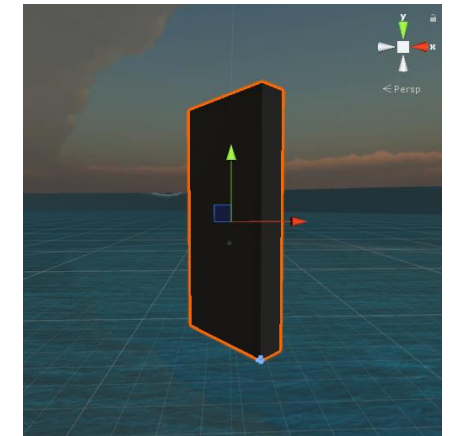
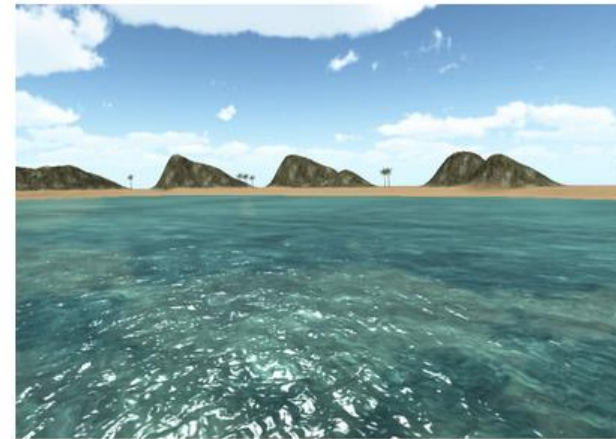
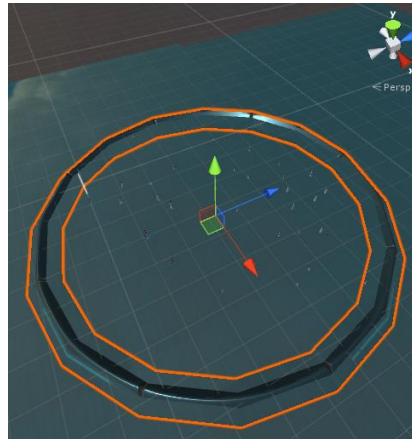
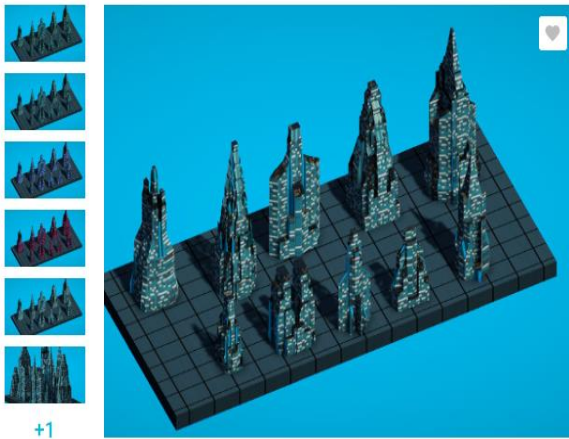
Implementation – Tools

- Software tools:
 - Game development: Unity 2017.3.1 [8]
 - Backend: C#
 - 3D modeling: Blender 2.79a [9]
 - Prototyping and textures: Adobe Photoshop CS6 [10]
- **Resources:** Intel(R) Core(TM) i7-6700HQ CPU, 2.60GHz, 8.0GB RAM, 1TB memory, NVIDIA GeForce GTX 960M, on Windows 10 OS



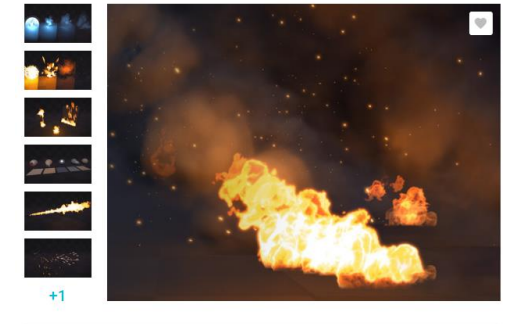
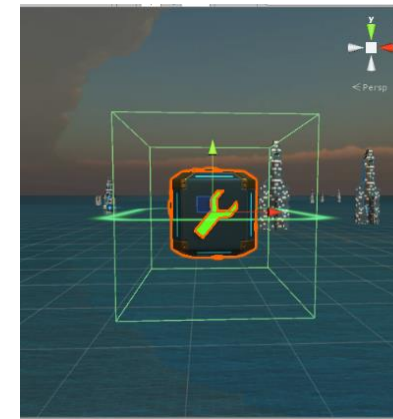
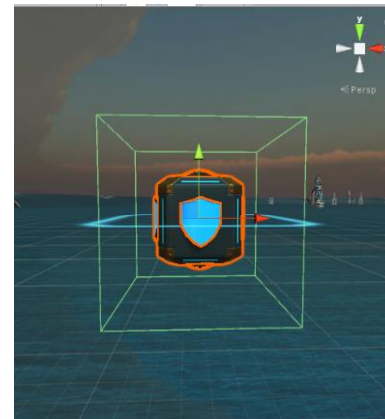
Implementation – Game Objects

- Static objects:
 - Buildings – for modeling the game scene
 - Delimiter ring
 - Terrain – water
 - Monolith



Implementation – Game Objects

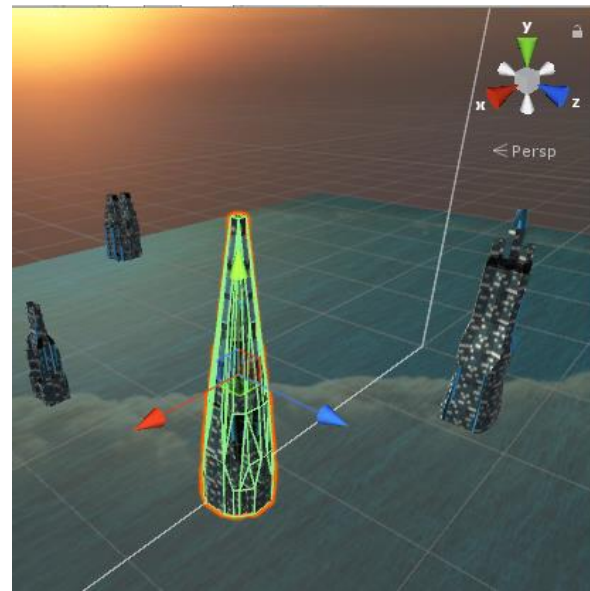
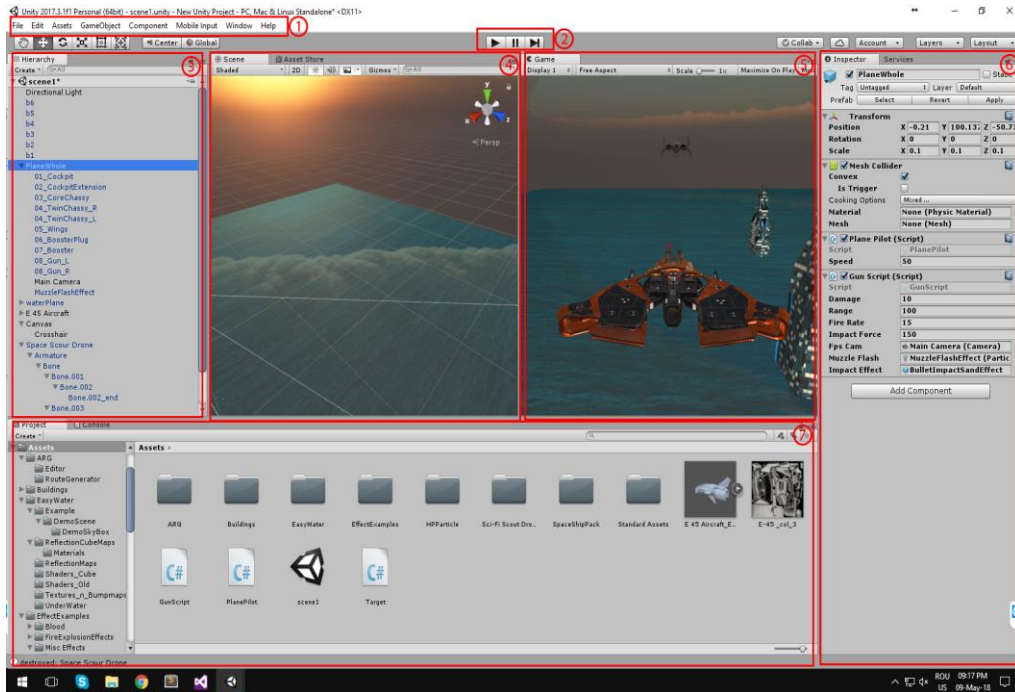
- Dynamic objects:
 - Flying car
 - Flying enemy vehicles
 - Power-up boxes
 - Particle effects



Implementation – Details

1. Game scene creation

- Import game objects and establish their position in the game world
- To buildings add rigid bodies and collision meshes
- Main camera attached to the flying vehicle of the player



Implementation – Details

2. Implement interactions between the player and the game

- Press Y key to start game
- Press ESC key to exit game
- Press Space to increase speed of the flying car
- Use Scroll Wheel to zoom in and out – changes the field of view of the camera
- Smooth the movement of the camera by adding physics
- The position of the car follows the mouse movements on the screen
- Establish the classes needed, together with attributes and methods (eg. VehiclePilot, Target, GunScript, RepairScript, ImmunityScript etc.)



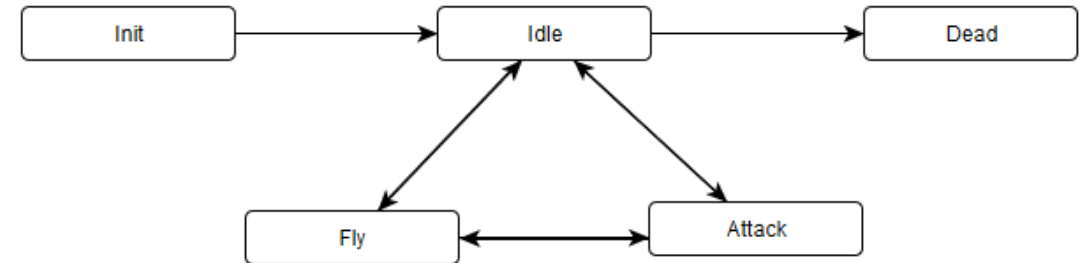
Implementation – Details

3. Implement interactions between the player and the car

- Method to attack that deals with attributes such as speed, health, and attached game objects
- Method to deal with being attacked by enemies (TakeDamage)
- On death the game is lost

4. Implement the enemies

- Artificial intelligence = Finite State Machine with 5 states
 - 1) Initial – the enemy is initialized at a random position in the game world
 - 2) Idle – the enemy checks the environment continuously, by rotating in a circle, around a pivot, but does nothing else;
 - 3) Fly – if the player is at a certain distance smaller than a set value, and if it is in its field of view (for example, 60 degrees), the enemy flies towards the spotted vehicle;
 - 4) Attack – and attacks;
 - 5) Die – if the amount of life reaches zero, the object is destroyed.



Implementation – Details

5. Implement the power-up boxes

- Immunity box – makes the enemy attacks null
- Health box – restores missing amount of life

6. HUD manager

- Deals with the display of messages that are triggered by different actions
- Contains all messages initially set as inactive
- Whenever an action happens, it sets the object active for a few seconds

7. Monolith

- Spawns at a location far away from the player
- If the player finds it and flies towards it, when it gets close enough to it the game is won

Evaluation

- Done after each of these stages, together with the documentation:
 - Analysis – task description and analysis, initial prototyping, game scenarios
 - Game design – 3D game scene, game strategy, interaction techniques
 - Game implementation
- Types:
 - **Functional** – done by the developer to test the implementation’s correctness
 - **Heuristic evaluation of usability** – done by evaluators based on Nielsen’s 10 usability heuristics for user interface design [11]

Evaluation – Summary

Usability level: 92.6%

1

Level of entertainment

Problem: small map, and easy to win

Solution: make the game scene bigger, and add more enemies, with more variety to their behavior

2

Errors

Problem: building collision

Solution: alter the bounce parameter of the physics materials

3

Difficulties

Problem: not enough sounds and visual effects

Solution: add a menu, new sounds particle effects, and life bar for enemies

Conclusions

- Team effort
- Requires time and patience
- Continuous evaluation is needed after each stage

- **Future improvements:**
 - Make the game scene bigger and more varied (randomly spawn the monolith, vary the enemies, add more power-up boxes, possibility of choosing vehicle or attack type etc.)
 - Introduce more sound and visual effects
 - Expand the story with different worlds
 - Multiplayer
 - Smartphone

Bibliography

1. K. Salen and E. Zimmerman, *Rules of Play: Game Design Fundamentals*: The MIT Press, 2003.
2. E. Adams, *Fundamentals of Game Design*: New Riders Publishing, 2009.
3. M. Sellers, "Designing the Experience of Interactive Play," *Playing video games: Motives, responses, consequences*, vol. Vorderer, P. & Bryant, J. Mahwah: Lawrence Erlbaum Associates, 2005.
4. MORS Special Meeting on Gaming, Background for the meeting in Fairfax, Virginia, 29 September to 1 October 2015.
5. A. C. Clarke and S. Kubrick, *2001: a space odyssey*, 1968.
6. D. Villeneuve, H. Fencher, M. Green, *Blade Runner 2049*: Warner Bros. Pictures, Sony Pictures Releasing, 2017.
7. C. Nolan and E. Thomas, "Dunkirk," 2017.
8. A. Okita, *Learning C# Programming with Unity 3D*: A. K. Peters, Ltd., 2014.
9. C. Wartmann, *The Blender Book: Free 3d Graphics Software for the Web and Video with Cdrom*: No Starch Press, 2000.
10. S. Onstott, *Adobe Photoshop CS6 Essentials*: SYBEX Inc., 2012.
11. J. Nielsen, *Usability Engineering*: Morgan Kaufmann Publishers Inc., 1993.

Game Development Methodology Mapped on the EvoGlimpse Video Game Experiment

Thank you!



Presenter: Zelia Blaga

Contact: Zelia.Blaga@cs.utcluj.ro

