

## Regulamento:

---

Nº de Alunos por grupo:	3 (Max)
Peso do Trabalho:	45% da nota final
Esforço estimado	25h de trabalho/estudante (o trabalho deve refletir o esforço esperado)
Data Limite para definição de temas:	30 de Abril de 2025
Data prevista de Entrega:	1 de Junho de 2025 (relatório + software com código fonte)
Oral (aleatório)	A definir
Nota importante:	A utilização de material já desenvolvido, ou tentativa de plágio implicará uma classificação final do trabalho de <b><u>zero valores</u></b> .

## Descrição do Trabalho:

---

Pretende-se que o grupo de trabalho desenvolva uma aplicação multimédia interativa sob a forma de um jogo de computador, sendo a utilização do elemento CANVAS HTML5 obrigatória. O tema do jogo é livre. É valorizada a utilização e programação de fenómenos físicos como aceleração, física e fricção, bem como técnicas avançadas de colisões, *scrolling*, otimizações de performance, *tiledmaps*, etc. O jogo deve ter pelo menos 2 níveis completos e funcionais.

É obrigatória a utilização da **metodologia adotada** e da **API** em desenvolvimento nas aulas da Unidade Curricular (Entidades, Gestor de Spritesheets, Gestor de Áudio, etc...) e da *framework* [extend.js](#).

Qualquer envio de mensagem para o utilizador deverá ser feito em HTML. O **uso do método `window.alert()` desconta 1 valor por cada linha** (10 `window.alert()` → `notaFinal-10`)

## Dicas para o desenvolvimento do trabalho:

---

O *design* de um jogo é um processo complexo que compreende várias tarefas. O resultado deste processo é um documento (*blueprint*) com especificações e requisitos, segundo os quais o jogo

deve ser construído (Ver anexo). Dado o tempo e recursos limitados, para a elaboração de um trabalho deste género, os grupos poderão organizar as tarefas da seguinte forma (método simplificado):

1. **Definir o tema a desenvolver** (importante)
2. **Produção de *Brainstorm* (chuva) de ideias.** O grupo deverá discutir e produzir várias ideias criativas relativas ao trabalho prático. (e.g., a utilização de mapas mentais, poderá ser uma ferramenta útil para estimular o pensamento radiante!
3. **Produção de *Storyboards/ Blueprints e Guião*.** Um *storyboard* é um rascunho inicial da peça que se quer produzir. Pode ser um rascunho em papel do trabalho a realizar, que traduz as ideias produzidas na etapa 2.  
**Nota:** Desta fase podem resultar vários *documentos*, dependendo das propostas alternativas que o grupo querará produzir para posterior escolha.
4. **Recolha de elementos gráficos e de imagem.** Após a etapa anterior o grupo tem já uma ideia definida do que pretende realizar. Nesta fase deverá pesquisar/produzir os elementos gráficos (e.g. spritesheets, imagens, etc...).
5. **Autoria** - Esta é então a fase final do trabalho. Com base nos requisitos definidos pelo *storyboard/Blueprint* e guião e nos elementos gráficos recolhidos/produzidos, o grupo deverá proceder à autoria da aplicação multimédia Interativa.

## **Critérios de Avaliação**

---

Na avaliação do trabalho prático serão tidos em conta os seguintes critérios:

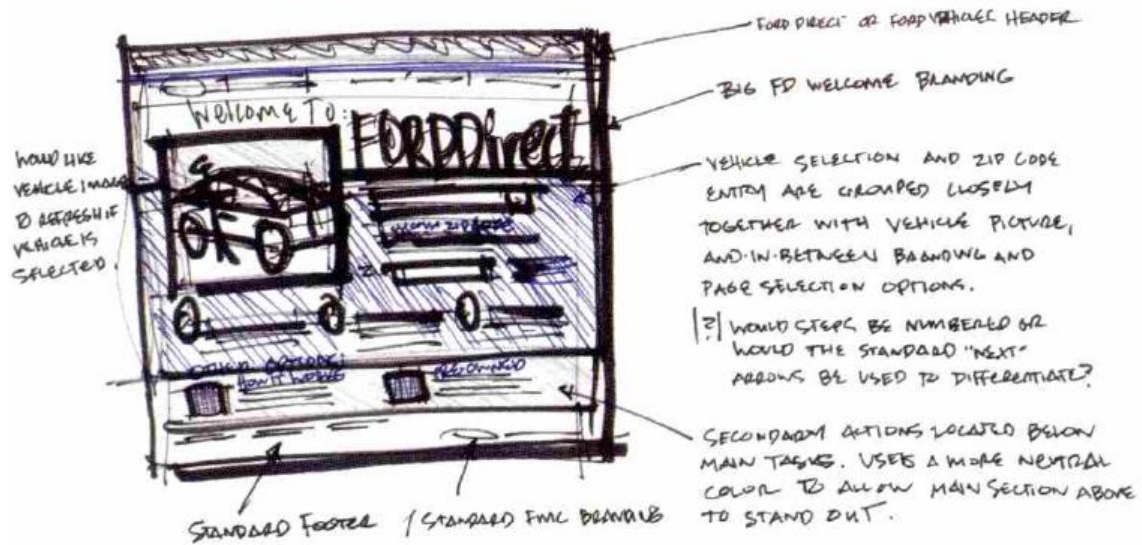
1. **Qualidade e complexidade da aplicação** (rigor)
2. **Implementação do motor de Jogo** (animações, interatividade, colisões, física, *scrolling*, etc).
3. **Gestão e manipulação de entidades** (Personagens, cenários e outros elementos)
4. **Boas práticas de programação em Javascript** (Orientação a Objetos é obrigatória).
5. **Riqueza dos conteúdos e respetiva manipulação** (Áudio, animações, etc.)
6. **Relatório final, segundo o modelo disponibilizado.**

## **Fontes de obtenção de imagem digital de qualidade, spritesheets e áudio:**

---

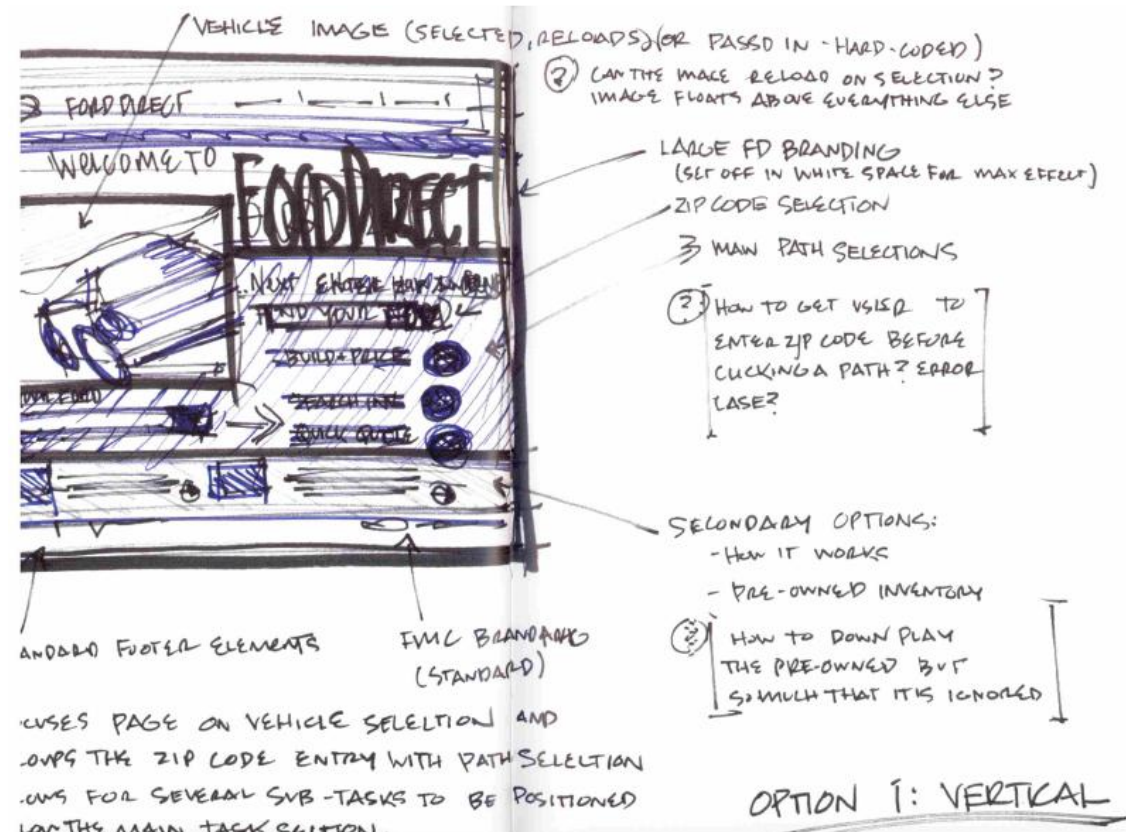
- Consultar a listas disponíveis no moodle da UC. Devem ser usadas spritesheets/tilesheets/audio de utilização livre (e.g., Creative Commons), e dados os respetivos créditos aos autores, numa janela/painel da aplicação com as respetivas hiperligações.

## ANEXO



POSSIBLY MORE SUCCESSFUL THAN THE VERTICAL APPROACH AS IT BRACKETS THE ZIP CODE ~~REQUEST~~ REQUEST IN BETWEEN THE VEHICLE SELECTION/DISPLAY AND THE ACTIONS TO LEAVE THE PAGE.

OPTION 2: HORIZONTAL



## Game Design Document Outline

North Carolina University, Department of Computer Science

A game design document is the blueprint from which a game is to be built. As such, every single detail necessary to build the game should be addressed. The larger the team and the longer the design and development cycle, the more critical is the need. For your purpose, the intent is to capture as much as possible of your design. I want you to think big...bigger than what you are able to develop. I also want you to be clear about what the software delivers and what the design entails. My recommendation is that you define the ultimate game and then clarify what it is that you have developed. If you are finding it too difficult to do that, you may produce too documents.

1. Title Page
  - 1.1. Game Name – Perhaps also add a subtitle or high concept sentence.
2. Game Overview
  - 2.1. Game Concept
  - 2.2. Genre
  - 2.3. Target Audience
  - 2.4. Game Flow Summary – How does the player move through the game. Both through framing interface and the game itself.
  - 2.5. Look and Feel – What is the basic look and feel of the game? What is the visual style?
3. Gameplay and Mechanics
  - 3.1. Gameplay
    - 3.1.1. Game Progression
    - 3.1.2. Mission/challenge Structure
    - 3.1.3. Puzzle Structure
    - 3.1.4. Objectives – What are the objectives of the game?
    - 3.1.5. Play Flow – How does the game flow for the game player

- 3.2. Mechanics – What are the rules to the game, both implicit and explicit. This is the model of the universe that the game works under. Think of it as a simulation of a world, how do all the pieces interact? This actually can be a very large section.
  - 3.2.1. Physics – How does the physical universe work?
  - 3.2.2. Movement in the game
  - 3.2.3. Objects – how to pick them up and move them
  - 3.2.4. Actions, including whatever switches and buttons are used, interacting with objects, and what means of communication are used
  - 3.2.5. Combat – If there is combat or even conflict, how is this specifically modeled?
  - 3.2.6. Economy – What is the economy of the game? How does it work?
  - 3.2.7. Screen Flow -- A graphical description of how each screen is related to every other and a description of the purpose of each screen.
- 3.3. Game Options – What are the options and how do they affect game play and mechanics?
- 3.4. Replaying and Saving
- 3.5. Cheats and Easter Eggs
4. Story, Setting and Character
  - 4.1. Story and Narrative – Includes back story, plot elements, game progression, and cut scenes. Cut scenes descriptions include the actors, the setting, and the storyboard or script.
  - 4.2. Game World
    - 4.2.1. General look and feel of world
    - 4.2.2. Areas, including the general description and physical characteristics as well as how it relates to the rest of the world (what levels use it, how it connects to other areas)
  - 4.3. Characters. Each character should include the back story, personality, appearance, animations, abilities, relevance to the story and relationship to other characters
5. Levels
  - 5.1. Levels. Each level should include a synopsis, the required introductory material (and how it is provided), the objectives, and the details of what happens in the level. Depending on the game, this may include the physical description of the map, the critical path that the player needs to take, and what encounters are important or incidental.
  - 5.2. Training Level
6. Interface
  - 6.1. Visual System. If you have a HUD, what is on it? What menus are you displaying? What is the camera model?
  - 6.2. Control System – How does the game player control the game? What are the specific commands?
  - 6.3. Audio, music, sound effects
  - 6.4. Help System
7. Artificial Intelligence
  - 7.1. Opponent and Enemy AI – The active opponent that plays against the game player and therefore requires strategic decision making
  - 7.2. Non-combat and Friendly Characters
  - 7.3. Support AI -- Player and Collision Detection, Pathfinding
8. Technical
  - 8.1. Target Hardware
  - 8.2. Development hardware and software, including Game Engine
  - 8.3. Network requirements
9. Game Art – Key assets, how they are being developed. Intended style.