

Project description

Web Games (VJ1217)

Academic Year 2023-24

Reference game: <https://www.crazygames.com/game/dino-defense>

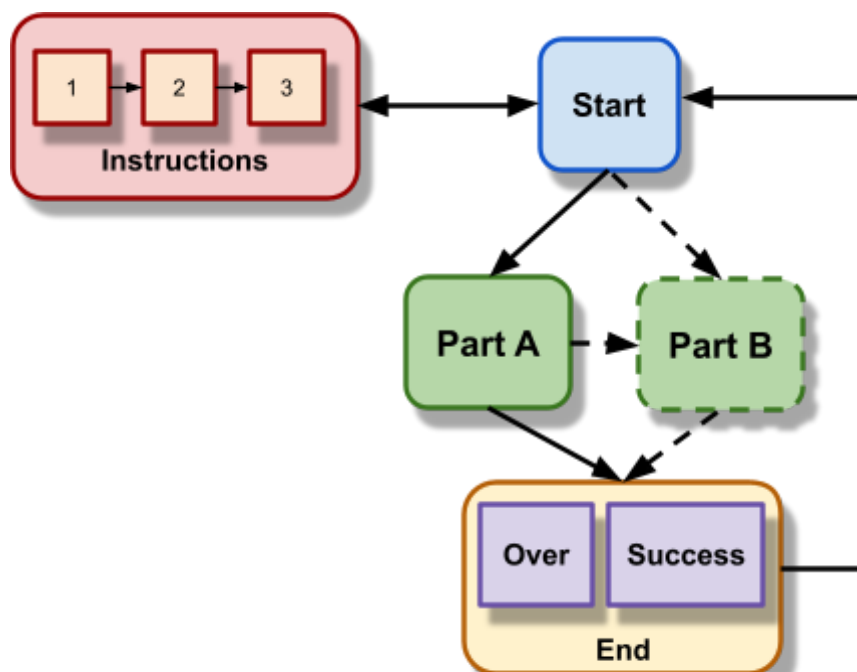
Required: Part A (grade up to 75%)

1. Start screen:
 - a. Game name and credits (names of team and team members).
 - b. Cool background image and, at least, one animation with a tween.
 - c. Selectable difficulty level (Easy, Medium, Hard).
 - d. Access to instructions screen (with a button).
 - e. Access to *any* implemented game part (Part A and Part B if done).
2. Instructions screen:
 - a. A set of “slides” describing the gameplay and game features.
 - b. Transition between slides with timer and button (for impatient players).
 - c. Return to the Start screen after the last slide.
3. HUD in-play screens (both for part A & B if done):
 - a. Life bar.
 - b. Score.
 - c. Difficulty level.
 - d. Remaining time in the safe area (see below).
 - e. Other relevant information such as the weapon (“firing tool” in general) or objects that the character has at any given time.
4. General gameplay:
 - a. Character moves with arrow keys (and/or A, S, W, D keys).
 - b. Character rotates around its centre according to her/his relative position with regard to the mouse coordinates.
 - c. Firing with mouse click: the orientation of the fired items matches that of the character.
 - d. World size at least twice the size of the stage (canvas): use Phaser’s camera model as described in P6.
 - e. Enemies move freely around.
 - f. When enemies get “close enough” to the character, they follow it (as the mice in P6), and exhibit a distinctive behaviour (different animation).
5. Additional details:
 - a. The world and its elements (positions) are described in a JSON file which is loaded (as in P6). The same happens with part B (if implemented).
 - b. ☒ Some world parts are locked by default and only get unlocked according to good performance (e.g. upon getting a particular powerup or passing a given score).
 - c. ☒ Two types of enemies with different behaviours/attacks.
 - d. ☒ Two types of objects to collect (e.g. money and gems, courses and degrees, books and prizes, depending on the theme/narrative behind your game).
6. Recharge areas
 - a. The character has limited “ammunition” to fire. When exhausted, mouse clicks will have no effect.

- b. There are several recharge areas around the world which the character should reach to recharge ammunition. When recharging the game pauses and an animation is played.
 - c. ■ The game is resumed after some event (e.g. animation is over, some time elapsed or a key is pressed).
- 7. Single safe area:
 - a. Enemies cannot get into the safe area (that's why it is safe).
 - b. The character can exchange some obtained objects (books, money...) for either some more powerful weapons, powers or life.
 - c. ■ The available weapons, powers or life items and their cost should be clearly depicted (e.g. machine gun = 100 coins, 10 life items per fruit item...). The objects should be subtracted and the new weapon/power/life added so that the HUD is properly updated.
 - d. The character can only stay in the safe area for a maximum of 10 seconds. When there are 3 seconds left, an alarm sound should warn the player about that.
 - e. If the character is not outside the safe area before the timeout, the life is exhausted (and the game is over).
 - f. A single exchange action can be performed per visit to the safe area.
 - g. ■ Outside the safe area, the player should be able to select the new weapon/power if it were the case (e.g. with specific keys).
- 8. End screen:
 - a. It has a button to return to the Start screen.
 - b. It has different contents for the failure or success cases.
 - c. Game over screen: if life is exhausted.
 - Final score and total number of objects collected.
 - d. Goal accomplished screen: upon a final special last part is unlocked.
 - Remaining life.
 - Final score and total number of objects collected.



Remember: Describe clearly in the project report (maybe also in 2a) your choices in parts marked with ■



Game screens. Pay attention to the navigation possibilities between them

Optional: Part B (up to the remaining 25%)

As “part B” we really mean the additional features that you can voluntarily implement to extend your game and thus be able to achieve the maximum mark (10). This can be achieved by:

- including a new part (“screen”) to continue your game with some new elements (e.g. more types of enemies) and some substantial and playable variation of the gameplay, while keeping the same general game theme and genre, or by
- implementing a new set of features or “behaviours” which enhances the actual gameplay required in part A.

Therefore, having a new part (continuation) of your game **is not strictly necessary**. Observe that in Figure **Game screens** part B box and the arrows related to it are dashed for this reason. In any case, the development effort to implement this extension of the game should be neither too low nor too high. As a rough reference, consider a little less than half of the effort required to implement part A as described above. Assume a linear relationship between effort and grade.

Below you will find some suggestions for possible features to implement to enhance part A and the corresponding mark that can be achieved if properly developed:

- To free/rescue “people” as in the reference game (four as maximum). When freed/rescued they must follow the character and they can be used as “slaves” to work and/or kill for her/him. If properly implemented this feature can give you a maximum of 2.5 points.
- To have a chance to exchange some of the objects collected for “pieces” that can be used to build walls or fences (“defences” in general) to protect the character from the attacks of enemies. If properly implemented this feature can give you a maximum of 1.5 points.
- To have “special” areas where you can get extra life, “objects” or “weapons” by playing a mini-game on another screen. This means that you will have to store the “state” of your game so that it can be properly resumed after completing the mini-game. If properly implemented this feature can give you a maximum of 2.5 points: 1 point for properly storing/resuming the game and between 0.5 and 1.5 points for the mini-game (depending on its complexity).
- To place special “boxes” in the game area which provide the character with random powerups such as invulnerability or undetectability for a limited time or with “super-weapons” or timed explosives which destroy all of the enemies within a given area. If properly implemented this feature can give you between 1.5 and 2.5 points depending on the complexity of the final powerups and/or weapons implemented.

If you opt for implementing other features to enhance part A or developing a continuation of the game (part B) then you will have to discuss your ideas with the lecturer supervising your work **prior to** the development itself. **The lecturer must be satisfied with your proposal:** he will give you feedback and a tentative mark (between 1 and 2.5 points) if considered adequate. Be creative!

Please, do not forget **to clearly describe** the new features and/or “behaviours” implemented to enhance part A **in the project report** if you finally decide not to explicitly develop a part B to continue your game.

Important

- The game must be implemented using Phaser-CE and HTML5/CSS/JavaScript.
- Refer to Figure **Game screens** and **the game specification** during development and writing up stages.
- All texts in the game should be written in correct English.

- At least one web font (e.g. from Google Fonts) should be used in some of the texts in the game.
- Transitions from/between game parts and screens should be smooth. You may use a Phaser state to implement these transitions.
- Animations and/or sounds should be used for these events at least:
 - Getting items.
 - Firing.
 - Killing enemies.
 - Entering/exiting the safe and recharge areas.
 - Exchanging objects.
- In order to facilitate the review procedure, please **make sure** that the submitted game fulfils the following aspects:
 - Each implemented part should be fully playable in **2-3 minutes**; they should not take longer than that. The **easy level** should be playable in about 1 minute.
 - Although the game challenge may increase from level easy to hard and part after part, make the **difficulty affordable** to be easily tested.
 - All required functionality which has been implemented **should be verifiable** in a single play of the game. This can be particularly important in those aspects regarding probabilities or frequencies.

Unimportant

- The exact appearances of the elements as in the game reference.
- The quality of art and aesthetics; do not spend too much time on them.

Procedure details

Teams

- Size: for proper task division 3 members are recommended (2 allowed only in special circumstances)
- Lab groups:
 - Preferably, members should be officially assigned to the same lab group.
 - If required for practical reasons, teams might be redistributed to lab groups.
 - Your work will be supervised and assessed by the lecturer of the lab group your team is assigned to.

Report

- You will be asked to submit a report, written in English, along with your game, in which you document things such as:
 - the software design decisions and implementation details, and the rationale behind them;
 - how you organised and coordinated the team activities;
 - how you managed time and how your project evolved.
- It is therefore advisable that you keep writing down important and relevant information. Some pieces can even become part of the final report. This all will help you remember and prepare the final document.

Submission deadlines

- May 30, 2024, 23:59.
- June 25, 2024, 23:59.